Chapter 5: The simulator

A general theory of the avatar cannot be directly applied to the domain of computer games. Computer game environments are algorithmic systems as well as formal game systems. The latter dimension is illustrated also by the example of table hockey as used above: what happens to the 'avatars' on the field does not make much difference unless it has some significance within the rule-system that defines what the game is about; therefore, if one of those miniature figures is accidentally knocked over, it takes a fair bit of added make-believe to make it into something more than merely an unfortunate break in the game). Moreover, the majority of computer games are *screen-based* media, which means that the avatar needs to 'translate', as it were, between the world of our bodies and a world of moving images. These questions will be addressed in this and the following chapter.

A computer simulation, according to the general definition outlined in chapter 2, is an implementation of a model which is not performed by a human participant, but by computers: the computer (or several computers) runs a simulation. Depending on the context, 'simulation' can also have more specific meanings, which I would argue are compatible with the general definition even if they have a different emphasis. In scientific, industrial and educational contexts, 'computer simulation' typically refers the activity of *modelling* for computer simulations. Roger D. Smith provides a concise and domain-independent definition of this concept:

> Simulation is the process of designing a model of a real or imagined system and conducting experiments with that model. The purpose of simulation experiments is to understand the behavior of the system or evaluate strategies for the operation of the system. (Smith 1999:2)

This definition does not necessarily exclude entertainment simulations, even if its emphasis is on modelling and simulation as an experimental and cognitive-analytical tool. Computer game theorist Gonzalo Frasca presents a similar, but broader definition when he says that simulation is "the modelling of a dynamic system through another system" (Frasca 2004:86).[47] This definition also refers primarily to the activity of constructing, implementing, manipulating and re-working formal systems.

However, in everyday language, 'a simulation' may also refer to what could be more precisely called a *simulator*: a simulating system considered as a self-contained machine; a machine that simulates. The concept of the simulator is most commonly associated with entertainment or training. It also captures, I want to suggest, the central fiction-making capacities of computer games, as these are manifested through software and hardware. Whereas non-computerised game systems are often also 'functional representations' or models, which can be implemented by players who take the role of 'simulators' (agents who perform a simulation), computer games are both models *and* simulators.

We can of course imagine many kinds of simulating machines that are not necessarily computers; a familiar example would be motion simulators, which essentially depend on video/film and various mechanical devices that are being synchronised with the images. Computer game simulators, however, are *cybernetic* machines; cybernetic simulators. The cybernetic simulator's essential capability is the automated implementation of algorithmic models. It is a procedural machine, a machine that can simulate processes all on its own – provided those processes are interpreted through abstract models that the machine is able to compute.

This is how Ted Friedman describes the cybernetic relationship between the computer and the player:

> What makes interaction with computers so powerfully absorbing – for better and worse – is the way computers can transform the exchange between reader and text into a feedback loop. Every response you make provokes a reaction from the computer, which leads to a new response, and so on, as the loop from the screen to your eyes to your fingers on the keyboard to the computer to the screen becomes a single cybernetic circuit. (Friedman 2002)

This notion of the cybernetic feedback loop or circuit has been theoretically formative within the emerging field of computer game studies[48]. The conceptualisation of computer games as cybernetic systems, introduced by Aarseth's concept of the 'cybertext' (1997), captures the dialogical relationship between the player

47 For the original version of Frasca's use of the term, see *Video Games of the Oppressed: Video Games as a Means for Critical Thinking and Debate* (Frasca 2001), where Frasca investigates how computer simulations can work as tools for role-playing with oppressive structures and mechanisms in society.

48 See Lahti (2003), Kücklich (2002), Dovey and Kennedy (2006), and Giddings (2006).

and the computer, a relationship in which the player is struggling to get the upper hand in a continuous exchange (or 'loop') of information and control[49].

Second-order simulations

A fully automated computer simulation only needs a set of initial conditions that defines for it a point of departure. Once kicked off, such a simulation works as a closed cybernetic feedback loop, a self-controlling mechanism that is running in dialogue with itself. The open or 'interactive' computer simulation, by contrast, requires the user to stay in the loop, as part of the machinery of simulation[50]. Together the user and the machine are bound together in a cybernetic dialogue, as reciprocal agents in a self-organising and self-controlling system.

However, there are two ways of staying in the loop; two different ideal models that define the role and nature of the user's (or in our case: the player's) participation with the process of simulation. These two ideal models describe how the player is positioned in relation to the activity of simulation, and define the modality of interaction that is available to the player.

In the standard mode of interactive computer simulation, the user participates as an equal partner in, or is in charge of, the process of simulation; the computer programme is a tool, an instrumental piece of technology that allows the player to perform simulations that otherwise would not be possible, or which would be significantly more laborious or impractical. The user (the scientist, the engineer, the student, the player) operates, manages and experiments with the process of simulation through observing results, varying input data, altering or tweaking the algorithmic models, and re-working underlying assumptions. The process of simulation is *transparent*, either because the programme is designed with a special-purpose interface that allows and facilitates transparency, or because the user is allowed to (and able to) change or modify the programme directly.

In contrast, in the non-transparent or *second-order* mode of interactive computer simulation, the user relates to the process of simulation only via the output produced by the simulation, with no access to the operations that produce the output. The user enters into a dialogue with the non-transparent

49 The principle of the feedback loop as part of a new discipline of 'cybernetics' was established by Norbert Wiener in *Cybernetics, or control and communication in the animal and machine* (Wiener 1948).

50 The notion of the 'interactive' simulation only makes sense in relation to a *computerised* simulation. All simulations are, by definition, 'interactive' (someone is simulating something), but because the computer has the unique ability to perform its own simulation, we often use the term 'interactive' when we refer to a computer simulation that includes a human participant in the loop even if it does not need to.

simulator according to the (often highly restricted) form and conventions that the programme dictates. The user is continuously influencing on how the process unfolds, but only indirectly, via the results as they are presented by the software and hardware. In the non-transparent computer simulation, the first-order simulation is transformed into an autonomous environment for the user to act within and affect upon.

A typical non-transparent computer simulation would be Joseph Weizenbaum's famous programme *Eliza* from 1964[51], a text-based chatterbot that simulates the responses of the stereotypical psychotherapist. Even if *Eliza* follows a set of fairly simple rules – she is basically responding to every input from the 'patient' with questions like "why do you say that x" or "tell me more about x" – , the simulation can still be convincing enough for the user to imagine that he or she is participating in a conversation with a psychotherapist. The user can only type text as input, and the programme can accommodate only a limited range of verbal inputs without making nonsensical responses.

In we are playing with Eliza, the computer takes the role of the simulator, by implementing the formal dynamic model that defines the behaviours of the stereotypical psychotherapist. This simulator is a mediator between the player and the 'rules' of Eliza; it accepts input from the player and feeds it back into the model, while keeping this model hidden from the 'patient'. From the point of view of the player, the actual workings of the model can not be accessed or observed, only inferred; after playing for a while, the player may figure it out, may decipher the code.

However, if the player is not given access to the rule-governed process that defines the simulation, in what sense can we still say that he or she still is performing or 'running' a simulation? How can a player implement rules without knowing them? The answer is that the player is engaging in a second-order process of simulation; this simulation implements the first-order simulation – the simulation that the simulator performs – as a *model*. The second-order simulation, performed by the user, implements the first-order simulation as a second-order dynamic model (a model of a model).

A good example of a second-order playable model would be Sony's robot dog *Aibo*, the cybernetic toy. *Aibo* is a functional representation which is itself already an implementation of an abstract model. Together, *Aibo* and the player form a second-order cybernetic system. The central difference between *Aibo* and computer games is that the former is neither a game nor a world, but a toy, a distinction I will return to below. Still, most computer games can also be described as second-order models. The first-order simulation that the computer performs is made dependent on the second-order simulation that the computer and the user

51 Eliza is currently available for consultation at http://www-ai.ijs.si/eliza/eliza.html

perform in dialogue. If the user gives no input, the first-order simulation will go idle, repeat itself endlessly, or simply just stop.

The loop of communion

As I noted above, the idea that computer game play is a cybernetic loop between the player and the game, as proposed by Aarseth (1997) and Friedman (2002), has been influential in contemporary new media and computer game theory[52]. This is how Friedman describes the experience of playing *SimCity* (Maxis Software 1989):

> It's very hard to describe what it feels like when you're "lost" inside a computer game, precisely because at that moment your sense of self has been fundamentally transformed. Flowing through a continuous series of decisions made almost automatically, hardly aware of the passage of time, you form a symbiotic circuit with the computer, a version of the cyborgian consciousness described by Donna Haraway in her influential "Manifesto for Cyborgs". The computer comes to feel like an organic extension of your consciousness, and you may feel like an extension of the computer itself. (Friedman 2002:5)

The use of the phenomenological idea of 'organic extension' has similarities with my analysis of the avatarial relationship above, but with one central difference: Friedman's 'complete communion' of absorbing experience, which is offered to the competent player (2002:4), is a communion with the *computer*, not with the avatar, and certainly not with a fictional world. While the notion of 'cyborgian consciousness' is linked to the specificity of the management- and strategy game genre, his account also echoes Sherry Turkle's classical study of arcade game players from the early eighties:

> People who have never played video games often think that success at them is like winning at a Las Vegas–style "one-arm bandit"; people who have played one game and given up acknowledge that they require "hand–eye coordination," often adding that this is something that children, but not they, possess. But success at video games involves much more. Working out your game strategy involves a process of deciphering the logic of the game, of understanding the intent of the game's designer, of achieving a "meeting of the minds" with the program. The video games reflect the computer within – in their animated graphics, in the

52 This theoretical model also connects game theory to a broader strand of cultural theory that is centred around notions of cyborgian or 'posthuman' forms of interaction, identity and politics (Haraway 1991; Hayles 1999).

> rhythm they impose, in the kind of strategic thinking that they require. This "computational specificity" becomes clear when you contrast the games with their "grandparent," pinball. (Turkle 1984:68)

Although describing two different genres of computer game play, both Friedman and Turkle describe a mode of experience according to which the player gets to merge with the logic of the computer, in a 'meeting of minds' between the player and the programme. Both are classical accounts of what computer gaming is about, and they capture the role of the computer in game play (the 'computational specificity') with a precision and analytical power that is lacking from more general accounts of digital media. At the same time, I would argue that both accounts reflect very *particular* paradigms of gaming experience. When taking a broader range of genres into account, these paradigms only cover part of the picture, especially when we consider avatar-based games.

The system simulator

Friedman's 'cyborgian consciousness' and Turkle's 'deciphering' articulate a particular kind of 'symbiotic' relationship that emerge from computer game play, accounting for how players, through hard learning and struggle, get into the cybernetic loop of mastery and control. Unlike the avatarial extension or prosthesis as described above, this relationship is all about getting into the guts of the machine, into 'the computer within', in order to be able to know it, to control it, to think like it, to become one with it in play. At the same time, *SimCity* and arcade games are also very different from each other; one is slow-burning and intellectual, the other frantic and tactical-visceral. While the former 'logic' is the logic of system dynamics, which invites the player to manage the parameters of change, the latter is all about pace, repetition and rhythm; in general, arcade action games are more about pattern than structure, emphasising variation over a theme rather than how a system evolves over time.

Also, we should note that only Friedman's account addresses directly the dimension of simulation. His focus is on the capacities of the system *simulator*, and on how it invites players to get under the hood of the on-screen simulation; the system simulator, in order to be mastered and conquered, requires the competent player to get at the constructedness of the simulated world, in a process of 'demystification':

> In fact, I would argue that computer games reveal their own constructedness to a much greater extent than more traditional texts. Pournelle asks that designers open up their programs, so that gamers can "know what the inner relationships are."

> But this is exactly what the process of computer game playing reveals. Learning and winning (or, in the case of a non-competitive "software toy," "reaching one's goals at") a computer game is a process of demystification: one succeeds by discovering how the software is put together. (Friedman 2002:3)

This description has similarities with what I referred to above as the standard or 'transparent' mode of computer simulation. Rather than engaging with the computer simulation as a second-order model only, the player is taking the position of the computer, identifying with the simulator itself, and is in this way relating directly to the procedures that govern the actions of the simulator. From this emerges a *reflexive* cybernetic feedback loop; the player becomes hyper-aware of the computerised specificity of the simulation, and the player's mind is able to tune in to the workings of the underlying formal structure. In a phenomenological sense, the competent player of *SimCity* inhabits the ecology of an abstract environment. Getting into the flow of playing the system simulator means *becoming* a system simulator. Being in the loop is to play the system. *Civilization* or *SimCity*, the strategy game genre and the simulation game genre, are paradigmatic models for this kind of play. The latter is also often called 'sim games' or management games.

Friedman's model of computer game play, more generally, implies a notion of computer game representation and computer game space that could be seen as the antithesis to the avatar-based approach that I have outlined in the chapter above. Considered as a model-based or 'procedural' representation, the simulated world of *SimCity* does represent (some aspect of) the real world, but more importantly: the miniature buildings, roads and parks also represent the system itself, the inner workings of the machine, and it is this 'inner reality' that the player has to grasp if he or she wants to get into the flow of the game. Consequently, when the player engages fluently in a transparent and 'demystifying' cybernetic feedback loop, the screen-simulated space of *SimCity* takes the role of an *interface* to the real workings of the game. The player does not use the machine in order to play with some domain in the world, but uses instead this domain – as an interface metaphor – to play with the machine and the programme. From the point of view of the player, the simulated environment that is represented on the screen may still be seen as a functional representation of something in the world, but only indirectly; only via the abstract model that it implements.

The 'world' of the game

Based on the principle of the second-order cybernetic model – the model that also performs a simulation – the simulator is able to produce a 'worldness' that sets it apart from other technologies of play and gaming. This worldness works against the deciphering imperative of system-oriented play. Marie-Laure Ryan's brief discussion of contemporary games in *Narrative as Virtual Reality* may serve as a cue to what kind of worldness we are talking about:

> In an abstract sense, of course, most if not all games create a "game world", or self-enclosed playing space, and the passion that the player brings to the game may be regarded as immersion in this game-world. But I would like to draw a distinction between "world" as a set of rules and tokens, and "world" as imaginary space, furnished with individuated objects. The pieces of a chess game may be labelled king, queen, bishop or knight, but chess players do not relate to them as fictional persons, nor do they imagine a royal court, a castle, an army, and a war between rival kingdoms. (Ryan 2001:307)

The fact that one does not normally engage with the fictional worlds of board games in the same way that one engages with the fictional world of *Tomb Raider* (Core Design 1996) or *Unreal* (Digital Extremes/Epic 1998), however, is not primarily due to the immersive power of "the sensorial representation of the gameworld", which is the dimension that Ryan chooses to emphasise (2001:308). The core difference between a world of 'rules and tokens' and a world of 'individuated objects' – which is a very central distinction – does not mainly have to do with the level of abstraction as such, understood as the complexity and richness of computer game imagery. The heart of the matter, I will argue, is the computer's capacity to implement formal models *for us*, so that we do not have to. As players, we are not asked to engage directly with the level of instructions, as we would have to do in a mimetic board game, where we would be required to do all the first-order simulation ourselves. Nor does a second-order model necessarily need to *reveal* the non-ambiguous and consistent game rules that govern it, and which could be deciphered and internalized by a 'demystifying' player as a world of rules and tokens.

3 principles of realistic agency

In contemporary computer games, the prime strategy for securing non-transparency and worldness, although not the only one available, is restrictive embodiment through the avatar. This embodiment is dependent on the simulator's capacity to simulate realistic agency. Let me suggest that realistic agency in simulated environments is premised on 3 general principles: *integration*, *reification* and *concretisation*.

The two different activities of simulation and game-play can only be seen as parallel in so far as they embody the process of *integration*, which means that the rules of the game melt with or are colonised by the rules that define possible actions, events and behaviours in a simulated world. This process is not the exclusive domain of computer simulations or computer games; colonisation is a principle at the level of instructions, and is therefore not dependent on a particular technology of implementation. Any game that also wants to be a simulation needs some level of integration. Chess is a simulation to the extent that some of the rules of the game can also make sense considered as principles of simulated warfare; we may note, for example, that pawns can only move forward and are not allowed to retreat.

Role-playing games, as well as war games and similar types of formalised simulations, are the paradigmatic form of non-computerised integration. In his paper "Word and code: code as world" (2003), Daniel Pargman accurately sums up how role-playing integration between game rules and world rules works – in a synthesis of formalised fiction that may seem peculiar and exotic to an outsider:

> The characteristics of different types of creatures, such as elves, dwarfs, dragons, ogres etc. are explicated, and their game-related behaviors and effects are laid out in great detail. The same is true for armor, weapons, magic spells and potions. The same is true for character traits and professions. And mental health. And equipment. And divine intervention. And so on. It is exactly such expositions – collected in thick rulebooks – which one gets hold of when a roleplaying game is bought. What is bought is a "game system" i.e. an operationalized system for how a (fantasy) world works in detail. (Pargman 2003:2)

Pargman's notion of 'code as world' describes the unique nature of the worlds of role-playing games as opposed to literary or cinematic worlds. Through the principle of integration, role-playing worlds become world-*systems,* which are highly complex yet 'logical and controllable' (Pargman 2003:1).

Reification is the kind of implementation that makes a computer simulation non-transparent. It is the principle by which the cybernetic simulator turns instructions into regularities, and abstract models into concrete models. This is possible because the simulator automates the execution of algorithmic instructions and keeps them hidden from the player. While the algorithms of the programme code are instructions from the point of view of the digital machine (which executes them), from the point of view of the human player they are simply a set of regularities, a cluster of (hopefully) consistent responses and behaviours. In a non-automated interactive simulation, by contrast, the player's interaction must follow the procedures as stated by the instructions of the system, without the mediation of a simulating machine. The procedures and behaviours that follow from instruction-based play can of course hold various degrees of realism in terms of how they relate to events and phenomena in the real world. However the interaction *itself*, the mode of interaction between the user and the system, will not be realistic in any sense.

By implementing a set of formal rules as a dynamic simulation of automated behaviours and appearances, the simulator puts flesh and bones on abstract models, potentially disguising their true origin and inner structure; the concrete *Aibo* implements the abstract *Aibo*. In computer games, instructions are reified as non-transparent objects and processes, which enter into relationships with other objects and processes in an autonomous and responsive environment, whose properties and behaviours embody 'rules' only in a metaphorical or heuristic sense.

Robots like *Aibo* also illustrate the simulator's capacity to simulate independent and autonomous *agency*, taking the position of an intentional subject or 'other' in relation to the player. This agency is rooted in the automated implementation of algorithmic procedures, whose quintessential generic form is if-then. Simulated agency adds an important dimension of realistic agency to our relationship with computer-generated environments: there are other intentional behaviours than our own. When playing a singleplayer computer game, there is *someone* who acts in relation to the player, a partner and opponent who is typically instantiated and 'localised' in various simulated agents and forces of the environment, but who may also – in different forms and to different extents – penetrate or animate the simulated world as a whole, which is a dimension I will return to in chapter 6. What I want to emphasise here is that simulated others – especially the kind of simulated agents that populate computer game worlds – are premised on a certain extent of reification and non-transparency. As humans, we do not have direct access to the operations or governing 'rules' of other independently deliberating minds. We can only know the intentions, preferences and strategies of other subjects through their actions, through interpretation.

The principle of reification is specific to the domain of simulation; it applies only to game rules that are integrated with or 'colonised' by a simulated system. A game system that does not model anything cannot be reified, only automated. This distinguishes reification from the computer's capacity to *enforce* or uphold the rules of a game. For example, in a racing game, the computer may force the player to follow a rule: the player/driver must stop when the time is up. This rule is not integrated with the rules that describe the properties of the simulated environment; no part of the simulation indicates that the vehicle should magically stop functioning after a set number of minutes and seconds, but such are the rules of the game, and the computer upholds them as unavoidable and non-negotiable absolutes. This rule is an explicit game-rule, an instruction, which is enforced by the computer much like a referee enforces the rules of football. In the case of sport games, the rules that state when a ball is out of bounds are admittedly more 'integrated' in the sense that they form part of a simulated game-space, but their central defining feature, with respect to reification, is nevertheless that they are explicit about their status as game rules. Automated sport game rules, just like the non-integrated rules that may cut your fuel when your time is up in a racing game, are not laws or properties, but 'regularities' that retain and speak openly of their conventionality and contingency as game rules.

Computerised chess illustrates a different kind of non-reified automated rules. When two participants play against each other, the computer enforces a set of rules that are also, in some respect, integrated, describing a military battle. However, the high level of abstraction inhibits reification by keeping the game rules unavoidably und unambiguously transparent, and by relying on the rule-governed behaviours of *tokens* rather than objects or agents to which we could ascribe believable properties. The simulation that is performed by the computer does not do anything that the simulation performed by the players in non-computerised chess cannot do. Because the automation of the computer in this case is essentially superfluous to the game as well as to the simulation (although convenient and efficient), computerised chess is most accurately called just that: a computerised game, not a computer game, incapable of mediating the process of reification.

In a similar fashion, although in far more elaborate and complex ways, computer role-playing games mitigate the principle of reification by keeping the rule-sets (which are often inherited from pen and paper versions) visible. This principle or ethos of anti-reification could be seen as a defining imperative of the role-playing game genre: the imperative to keep the world-system transparent; to keep the computer as a modest 'computer' in the original meaning of the term, doing the maths for you, and upholding the rules. We can therefore consider role-playing games like *Planescape Torment* (Black Isles Studios 1999) or *Neverwinter Nights* (Bioware 2002) as hybrids: part computer game, part computerised game – or alternatively: part computer simulation, part computerised simulation.

The principle of *concretisation* has to do with the function of concrete models within game systems. It points to a distinction which is either ignored or not given much importance in formalist game theory. Whereas pre-digital games typically use concrete models as *tokens*, a computer game typically uses concrete models as *playthings*. This means, for example, that computer games are well suited to simulated sport games like football or Formula One racing.

Seen from the point of view of the game system, a plaything is a *found object* (Juul 2005:67). A found object, in this perspective, is a game component that generates procedures of play through its own properties as an object, by virtue of its consistency and regularity of appearance and behaviour. For example, a football that does not have the correct shape and consistence, or a Ken doll that lacks his head, would seriously change (and potentially ruin) the activity of play. A token, in contrast, only needs to have (or to retain) very basic physical properties.

Sport and other physical games generally include both formal rules and playthings. In certain types of physical games, these playthings may also function as models, as mimetic toys, but this is generally the exception to the rule. An example that comes to mind is paintball, either as can it be observed in leisure centres, or as the televised sport where serious competitors dress up in military camo gear and shoot paint-splashing pellets at each other. However, as a general rule, mimetic toys –playthings used as models – generally tend to *compete* with or displace formal rule-sets rather than accommodate to them; we may engage in formally structured games with our Legos or our paper dolls, but their capacities as playthings tend to discourage it.

Board games, on the other hand, typically use concrete models as central components within the game system, but then those models are *not* (in ordinary types of play) treated as playthings or found objects. The properties of the miniature motorcycle model that we move around the board in Monopoly are not important in the same way that the properties of a football are important. The properties of the motorcycle matter only in terms of how they enable it to function as a token: the small figure has concrete mass, it is of proper scale; it cannot be in two different places at the same time. Given that such minimum requirements are met, the role and function of a token, by definition, is described by the formal instructions of the game system. These rules cannot operate if they are not understood and implemented by a player.

In contrast, if we are playing with a miniature vehicle as a toy – that is, outside the boundaries of any formal game system – its specific properties will be crucial in defining how we are able to play and what the playing means to us; playing with a wooden toy truck with painted-on wheels and playing with a radio-controlled plastic wonder are two different things (in spite of what our parents might want us to believe). In Monopoly, it does not matter if the little vehicle has painted-on wheels, or whether it represents a motorcycle or a sandwich or a cow. A token is by definition themable: it can take on any kind of appearance and still perform the same function within the rules of the game. The behaviours that can be ascribed to the properties of the motorcycle figure itself can never claim authority over the instructions that govern the activity of the game; if the miniature accidentally slides out of its assigned position or is knocked over, this behaviour is irrelevant to the state of the game. The properties of a token do not have the authority to generate unforeseeable actions and events within the game system. A plaything, on the other hand, like gravity or the human body, is a found object, which is interesting precisely by virtue of its capacity to 'instruct' or generate procedures of play.

A token can never have the same fictional significance as a plaything, because unlike a plaything, it cannot function as a model; as players we do not allow it to generate fictional actions and events by virtue of its properties as a functional representation. It may of course be used as a prop in a game of make-believe, but only in terms of its properties as a visual representation rather than as a functional representation – that is: in terms of its potential as a *depiction*, generating what Walton calls a visual (or perceptual) game of make-believe. Unlike the fictions generated by of mimetic playthings, therefore, the fictional dimension of board games is an add-on, a depictive overlay to the rule-governed procedures of play.

Computer game environments, unlike the gaming environments of Risk, Monopoly or pen and paper role-playing games, are all about the playthings. However, because these playable or 'found' environments are able to integrate the rules of the game, they do not compete with the requirements of formal game systems, as mimetic playthings usually do. This is a unique and revolutionary capability of computer games: *they use concrete models to colonise the rules of play*, and they can do so because those concrete models are implementations of dynamic abstract models; they are the result of reification. The simulated worlds of pen and paper role-playing games, in contrast, also colonise game rules, but not it via concrete models (– if they use concrete models, these are assigned the function of tokens). In computer games, the properties of the concrete, playable and generative toy are able to absorb and concretise the workings of the game system. The principle of concretisation, consequently, may also serve to distinguish computer games from computerised games: the latter emphasise the importance of tokens over playthings.

The principles of integration, reification and concretisation – the latter being a combination of the former two – explain how the simulator is able to offer a relationship to a simulated system that mimics our relationship to the real world. Realistic agency is when you do not have to perform the simulation by following a set of instructions, and when the behaviours of agents, objects and processes in the environment can be ascribed to their own properties and capabilities rather than to formal procedures that are external to them. In computer game environments, this kind of realistic agency is often combined with and balanced by game rules that are *not* concretised – either because they are transparent and non-reified (as in a role-playing game), or because they were never integrated with the simulation in the first place (as when a timer cuts you off in a racing game).

So the simulated environments of computer games are 'worlds' not only because they can trigger our imaginations, or because they constitute a rule-based and self-contained 'magic circle' of meaningful activity, *or* because they may be sensorially immersive, but also, and more importantly, because they are world-like in terms of our mode of interacting with them. Unlike non-computerised systems – whether these are simulations, game systems, or a combination of the two – we can interact with computer-simulated environments in a way that is analogous to how we interact with the world outside the simulation.

Finally, it is useful to point out the distinction between realistic agency, which is a particular property of computer-simulated environments, and the notion of functional realism, which has been suggested by Geoff King in his analysis of *Full Spectrum Warrior (Pandemic Studios* 2004), an hybridised action/strategy game that is marketed as a realistic and authentic military entertainment simulator. The functional realism of this game, King says, operates at the level of military tactics. Compared to other military-style shooters, the manner in which the player is forced to perform the basic tasks in *Full Spectrum Warrior* – the game's 'core mechanics' – corresponds more closely (or, we could add, less badly) to the way professional soldiers are actually trained to do combat in those kinds of environments (King 2005). This type of realism, I will suggest, does not primarily address the world-like responses of simulated environments, but a (presumed) homology at the structural level, a *functional* homology that can be expressed in entirely abstract terms, that is, in terms of the rules that govern the possible actions of the player. Functional realism is therefore distinct from realistic agency. As noted above, any simulation, for example a pen and paper simulation or a board game, can be measured and found realistic in terms of how its rules correspond to the perceived patterns and regularities of the particular domain that it simulates. Conversely, a computer game that offers a high degree of realistic agency and 'worldness' – say, for example, *Black* (Criterion 2006) – may not score very high in terms of functional realism.

Environments versus automatons

As noted above, mimetic toys do not generally mix well with formal game systems, as their capacity for generating procedures of play competes with and easily disrupts the authority of the rules. In contrast, computer games are concrete models *and* formal systems, or more accurately: formal systems as concrete models. This makes them more similar to intelligent toys and robots, to cybernetic *automatons* than to paper dolls or Lego men. A cybernetic automaton, like *Aibo*, implements a formal structure that defines its dynamic responses and behaviours as a concrete model. Because *Aibo* is a second-order model, we may, in certain respects, interact with it in ways that are analogous to how we would interact with a (slightly confused) puppy.

Whereas automatons do have the capacity to integrate game rules, however, we should note that in terms of fictional participation, they engage us as *agents* rather than as worlds. The difference between cybernetic automatons and cybernetic worlds can be described via Kendall Walton's notion of the 'work world'. Certain kinds of props – we may refer to them as 'world-props' rather than merely as props – generate a world of their own, and they do so in an exclusive sense; they cannot enter into a world of make-believe as one prop among others, because they are not reflexive with respect to their environment. A world-prop is a self-contained prop, a game of make-believe incarnated as a prop.

Typically, as in the case of paintings, books or films, world-props are meant to be used in perceptual games of make-believe rather than in model-based games of make-believe; we do not normally use them as dynamically reflexive props, props that makes fictional the changes that we effect in them. To the extent that it *is* possible to appropriate a novel or a painted portrait as a model – say, by burning the novel or throwing rotten eggs on the portrait – we are stepping out of their work-worlds in order to engage in a different game of make-believe, and so they are no longer world-props. Because neither novels nor portraits model environments, the model and the work world become incompatible; if we emphasise the model, we lose the work world, and vice versa.

On the other hand, this conditional incompatibility may also give us a flexible but yet reasonably precise definition of what a simulated environment is: a simulated environment – any simulated environment – is a model that is also a world-prop. While some simulated environments are built on abstract models, like card and dice simulation games, others rely on concrete models, like film sets or other kinds of mock-up streets, buildings or towns. A computer-simulated environment is in a way a combination of those two types of environments. Because it relies on reified algorithmic models, it is both informational and concrete at the same time.

The concept of the world-prop also serves to differentiate between two ideal types of cybernetic fiction, two types of props that are both informational and concrete. Whereas the cybernetic automaton, when used in games of make-believe, communicates with and 'fictionalises' its environments, the cybernetic world (or 'work world') offers instead a self-contained and sovereign simulated environment.

We may assume that cybernetic worlds are simply a product of screen-based simulations, and that the boundary of the screen constitutes the boundary of the fictional world. However, while this is often the case, it is not necessarily so. Some computer-simulated environments combine screen-projected and physical props. Flight simulators, with their elaborate full-size cockpit models, would be a typical example. Conversely, automatons may also be screen-based in different ways. *Eliza*, to illustrate, in spite of being screen-based, lends itself well to being appropriated as an automaton. Let us say that we are playing along in a game of make-believe, according to which there is an Eliza the psychotherapist typing to us from some other terminal, or magically residing within the computer, or whatever setup will make sense to us according to the situation. In principle, considered as a concrete model, the *Eliza* programme is then reflexive with respect to the physical environment that the player uses for the game of make-believe. The boundary of this fictionalised environment is not incarnated by or clearly defined by the prop itself; the screen or the keyboard may be a part of the environment, but what about the chair? The desk? In this case, the boundaries of fictionally relevant space are not made explicit or clear (– indeed, addressing or questioning them in the first place could be seen as nonsensical) because the *Eliza* programme is being used as an automaton.

Following a similar principle, screen-based computer games may, to a greater or lesser extent, draw on the model of the automaton in the way they appeal to the player's fictional participation. Examples would be *Nintendogs* (Nintendo 2005) and similar types of Tamagochi-inspired games, in which the player interacts with virtual pets as if they were a part of his or her everyday space and everyday life. Some games break out of their self-contained worldness primarily by absorbing (or invading) the player's world in a temporal more than a spatial sense. In *Animal Crossing* (Nintendo 2004), the events of a persistent simulated environment follow in sync with days and seasons in real time, defining the game-space a parallel place just as much as a separate world.

Unlike *Eliza* and *Nintendogs*, many computer games are self-contained cybernetic fictions; they are unambiguously 'work worlds'. This implies, as I will return to in chapter 7, that they are related to the screen-projected work-worlds of film and animation in a way that automatons, even screen-based ones, are not. However, in film and animation, the ontological boundary of fiction – the 'fourth wall' that defines the fictional world – is usually closely associated with the boundary of the *recounted*, which is the boundary of a diegetic storyworld. In computer game work worlds, on the other hand, this diegetic dimension (in so far as there is one) is subordinated to the here-and-now of mimetic play. This means that computer game fiction, in principle, can more easily extend beyond the boundary of the screen, just like it does when playing with *Eliza*. Therefore, in computer games, the difference between an automaton and a self-contained environment is not necessarily clear-cut or unambiguous.

Computer game worlds are also, just like the automaton, self-operating intelligent machines. System simulators like *Sim City* or *The Sims*, as discussed above, are in one sense more similar to automatons than other kinds of games, because they are engaged with from the outside, as small totalities or organisms, as cybernetic toys. However, because they are world-props rather than agents, our fictional participation with them is different. Rather than re-positioning us in a game of make-believe, as the automaton does, they are more comparable to construction toys like Lego, with which we, as noted in chapter 3, typically participate through *instrumental* agency rather than fictional agency. In terms of our fictional participation, then, they are self-contained rather than dialogical. A system simulator does not generate fictional truths about its dialogue with the player, as *Eliza* or *Aibo* does, but it generates fictional truths about itself, about the state of its world as a self-contained entity. This is why the system simulator lends itself especially well to the 'process of demystification' that Friedman talks about. The dialogue between the player and the machine produces a fictional world, but that dialogue itself is not part of the world that it produces. We may still, though, engage in fictional participation from inside this self-contained world, but this participation will need to operate on the level of mental make-believe.

Computer game worldness

To sum up: Sherry Turkle and Ted Friedman describe in what way the notion of 'worldness' in computer games is a paradox; part of the pleasure is to play with this worldness itself, to get under the hood, to indulge in the paradox of the world-system. However, while this kind of 'communion' between the player and the system reflects, on the one hand, a general appeal that is specific to computerised media, the cultivation of this mode of interaction is also highly dependent on generic conventions. At the same time, the paradox itself is rooted in the computer's capacity as a simulating machine, a simulator, which facilitates realistic, world-like agency. It is this realism that is the more important specificity of computer games.

Realistic agency takes fictionality beyond the status of the representational 'theme' or overlay, and beyond the metaphorical 'world' of rules and tokens. This fictionality is rooted in the same basic premise of pretence or virtuality that carries fictional worlds in other media, but it draws on the generative power of concrete – or more to the point: concretised – models rather than the generative power of depictions or verbal props. Through the power of the simulator to execute and reify formal models, computer games, like other computer simulations, give players the ability to interact in a world-like manner with fictional objects. Computer games are playthings, and as such they are comparable to toys and cybernetic automatons. Yet most computer games are self-contained worlds rather than dialogical agents; as simulated environments, they are both work worlds and concrete models.

Finally, it must be emphasised that the worlds of computer games are a special *kind* of self-contained simulated environments; they are, uncompromisingly, games as well as worlds. The computer game simulation integrates and concretises the explicit game rules that govern the actions of the player. The world of mimetic playthings merges with the game rules that govern our actions – not as a 'special case' but by default. In computer games, game rules are colonised by fiction.