One Simulated Hand, Two Real Hands: Antisimulation and Phenomenological Correspondence in Videogame Control Schemes

Joseph Garvin

University of Bristol

Abstract

This paper sets out a framework for understanding the control schemes of computer games in terms of 'phenomenological correspondence'. The complexity of the simulated action corresponds with the complexity of the player's action on the controls. With this, the traditional or popular view of controls as either 'simulator' or 'arcade' controls makes sense as different forms, either 'completionist' or 'reductionist', of correspondence. As well, a third kind of control scheme can be made sense of, the 'antisimulation' seen in games like *QWOP.* These controls involve an 'excessive' phenomenological correspondence, where one simulated hand requires two player's hands to control.

Keywords

Control schemes; simulator; arcade; phenomenology

Press Start Volume 2 | Issue1 | 2015
ISSN: 2055-8198
URL: http://press-start.gla.ac.uk

The control scheme of a game is often defined in terms of 'arcade' or 'simulation' controls (Birnbaum, 2014; Live For Speed, 2014; Digital Combat Simulator, 2014). Though most often applied to driving, flying, and military games - games types traditionally associated with the 'simulator' - it is extendible beyond those to any game where a 'real' (or at least possible) action is represented. This paper will set out two related claims - first, that these two descriptions are not merely jargon or marketing terminology but instead identify a real difference in how the phenomenology of different games relates to the phenomenology of a bodily action, and then further that this phenomenological framing sets up a description of a third relationship between game-action and bodily action. This third relationship I will call 'antisimulation'.

Firstly, an analysis of 'arcade'. The arcade approach is that controls should enable the player. These control schemes simplify (or reduce) complex actions to a few, even a single, button push. In terms of the simplicity of control, endless runners such as *Canabalt* (Semi Secret Software, 2009) are the most extreme form, with other minimalist games like *One Finger Death Punch* (Silver Dollar Games, 2014) coming close. Endless runners, especially the side-on kind which *Canabalt* is credited with creating, typically use a single input, such as a tap on the touchscreen or a press of a spacebar, which corresponds to a single player-controlled action, jumping. *One Finger Death Punch* uses only two inputs, a left or a right directed tap or button press.

In terms of simplification of complex actions, shooter games such as *Call of Duty 4: Modern Warfare* (Infinity Ward, 2007) and driving games such as *Burnout: Paradise* (Criterion Games, 2008) stand out. For all of these, controls are simplified compared to the action represented. In *Burnout*, starting the car is the same as moving forward, gearboxes apparently do not exist, and repairs are instant. *Call of Duty* continues the tradition of standard shooter mechanics and controls by turning reloading, aiming, and changing weapon each into a single button press (Smith, 2014).

The essence of the arcade approach is that controls should enable the player. It is not accidental that many of the typical arcade-controls games I have mentioned are well known as power-fantasies. While not all games with arcade controls are power fantasies (*Gone Home* (The Fulbright Company, 2013) would be a game with arcade controls, and it is hard to argue that that game is a power fantasy), these kinds of controls suit that experience. Arcade controls are easy, direct, and, at least when done well, intuitive, which enables the sense of power found in games like *Call of Duty* and *Burnout: Paradise*.

'Simulation' controls, on the other hand, aim at a direct correspondence between simulated action and player input. The most obvious examples are those sold under the name 'simulator', such as flight sims in the mode of Microsoft Game Studio's *Flight Simulator X* (2006), and detailed

transport/driving simulators such as the bus-driving simulator, *OMSI* (Aerosoft, 2011). Another example, that I will go into in detail later, is the first person shooter/gun simulator *Receiver* (Wolfire Games, 2012). Each of these aims at a direct correspondence between an action in the game and a player's action on the controls. In the flight and transport simulator genres, this can extend to the point where the control mechanism itself, rather than controllers intended to be in some way universally applicable, become intensely specific and directly replicate the control systems of a car or plane. The most dramatic example would be the large-scale flight simulators, built to resemble a plane cockpit, referred to as a 'full flight simulator'.

Each of these terms has its origins in how developers and players describe them. Simple, easy to use controls are regularly referred to as "arcadey" (Birnbaum, 2014). One driving simulator developer dismisses less detailed controls as 'arcade modes', saying that in their game there are "no arcade modes [...] YOU have to do the driving" (Live For Speed, 2014). Complex controls are described by developers as simulator controls - and simulator is often used as an ideal to be worked towards, with the ultimate simulation being something which is truly 'authentic' (Digital Combat Simulator, 2014). This distinction has its roots mostly on the simulator side of the divide, though this is easily explained as the product of simulators being a minority or niche genre, and so in need of language that distinguishes themselves.

There is, however, a confusion that may arise from the use of the term "simulation", as I will be forced to use it in two distinct terms. One of them is the idea of a control scheme that aims at perfect accuracy to reality - the term that is contrasted with arcade. The other is any game that attempts to mimic or represent some action, and so applies equally well to a highly accurate simulator (in the first sense of the word) such as *OMSI* and a easy to use arcade game such as *Burnout: Paradise*. In each case, 'simulation' is a well-established term, and so to try and introduce a completely new term would be at best awkward, if not possibly more confusing. Instead, I will distinguish the two by referring to the first sense as 'simulator-control' and the second simply as 'simulator'.

Once there is the idea of both the action (walking, flying a plane, cutting up vegetables, etc.) and the idea of a simulation of that action, which is distinct from the simulated, then there is inevitably the question of what they share and what they do not share. In this paper, I will focus on the comparison between the phenomenology of an action and the phenomenology of a simulation of that action. This relationship is what I will call phenomenological correspondence, or simply correspondence.

The sense of phenomenology that I'm using is that used by Hubert Dreyfus in his paper 'The Current Relevance of Merleau-Ponty's Phenomenology of Embodiment' (1996) and by Daniel Chandler in his

article 'The Phenomenology of Writing by Hand' (1992). It is a phenomenology of physical bodies, one that focusses on the specific experience of a particular action. Dreyfus seeks to analyse, in depth, the behaviour of a tennis player, both at a novice level and at an expert level. Chandler seeks to describe in detail the experience of the physical act of writing. I am not describing in such detail the act of controlling a game, but that concept of the detailed description of a physical act is what is underlying my approach of comparing the complexity of the player's physical act and the complexity of the action in the game.

Simulator-control games and arcade-control games each have a different form correspondence. If we take the most extreme cases of each - on the arcade side, games such as *Canabalt* and *Call of Duty 4*, and on the simulator side Receiver and complex flight simulators such as *Flight Simulator X* - this distinction in correspondence will be clearest. An arcade game seeks to simplify the game experience relative to the simulated action. I will call this a reductionist correspondence, as it seeks to reduce the variety of motions and actions involved. A simulator-control game seeks to accurately represent the simulated action in the game. I will call this a completionist correspondence, as it seeks to have as complex and varied a set of motions and actions involved in controlling the game as there would be in controlling the real thing. That this is a phenomenological relationship is clear. A particular action (driving a car at various speeds, firing and reloading a gun) has a particular phenomenological experience. An arcade-control game seeks to have a simpler, more immediate experience. A simulator-control game seeks to have an equivalently complex experience, or in the cases of full simulation, a phenomenologically identical experience. However, excluding full simulations, both arcade-control and simulator-control games rely on ‘micro-mobilities’, that is small, primarily hand-based actions performed on some kind of control unit such as a console controller, a computer keyboard, or a smartphone touchscreen (Richardson, 2011, pp. 420-421). These micro-mobilities cannot normally correspond in type to the simulated action - a press of a key is not the push of an accelerator. Instead, when I describe a correspondence as reduced or complete I mean that the number, complexity and variety of actions are reduced or represented completely.

These two states of reduced and complete correspondence fit into Hubert Dreyfus’s description of physical expertise and how that alters the phenomenology of a particular action. He discusses how the very same action (swinging a tennis racket and hitting a tennis ball) feels different depending on the level of expertise. A beginner is necessarily aware of all the details, and over time and with a gain in expertise, is able to, in a phenomenological sense, ignore certain details.

‘To get the phenomenon in focus, consider a tennis swing. If one is a beginner or is off one's form one might find oneself making an effort to

keep one's eye on the ball, keep the racket perpendicular to the court, hit the ball squarely, etc. But if one is expert at the game [...] what is experienced is more like one's arm going up and its being drawn to the appropriate position, the racket forming the optimal angle with the court - an angle we need not even be aware of - all this so as to complete the gestalt made up of the court, one's running opponent, and the oncoming ball.' (Dreyfus, 1996)

Expertise has the effect of reducing the phenomenological complexity, allowing the tennis player to focus on being in the right place, striking the ball at the right angle, and in general to focus on the effects of the action, not the action itself. In the context of arcade- and simulation-controls, an arcade-controls game shortcuts around much of the expertise required, building that expertise into the control scheme itself. While a player of simulator games can over time gain a remarkable level of skill and so be able to carry out much the same actions within the game as a player of an arcade game, the simulator-control scheme is built with player expertise as something to be learned.

Another valuable analogy is Merleau-Ponty's description of sensory inputs as coming together like a "melody" (2002, p. 125). Merleau-Ponty uses this analogy for the tactile sense, but it is similarly possible to use it to describe a bodily action. A smoothly executed action, such as flipping a coin or filling a cup from a teapot, is a series of sub-actions, like active versions of Merleau-Ponty's "local impressions", which come together to form one action. A badly executed action fails to follow through smoothly, and so the melody doesn't work. An arcade game aims at simplifying the action of playing the melody, supporting the player. A simulator-control game aims to give the player all the tools to play the melody themselves, but also removes all the support that would prevent them from failing.

Here, I will go into detail on three games, two from the arcade-control side and one from the simulator-control side. The two arcade games are the endless runner *Canabalt* and the first person shooter *Call of Duty 4: Modern Warfare*. The simulator game is the first person shooter/gun simulator *Receiver*.

Canabalt is considered the original 'endless runner', a game where forward motion is automatic and the player controls only movement up and down, via jumping. The challenge of the game comes with the variety of obstacles that block the path, and which must be avoided through well timed jumps. As such, *Canabalt*, like most endless runners, can be played with a single key or exclusively though tapping on a touchscreen. It would be difficult to imagine a simpler control scheme. This sort of control has, in reference to a different but similar game, Dong Nguyen's *Flappy Bird*, has been described as an extreme form of "intra-ludic seriality", though in this paper I will chose to use the term 'repetition' in place of seriality (Heilmann, 2014). *Canabalt*'s controls

offer variety only in timing - how often does one jump, and for how long. The control itself is always the same button, and the action is always the same action, a jump. Repetition is the core of the controls, and as such are extremely simple - Heilmann describes both *Flappy Bird* and *Canabalt* as games where "[o]ne single interface action [...] causes one single in-game action", and so as games that are strongly serial/repetitive, and hence very simple in their physical input (2014, p. 39).

While the specific details of *Canabalt*'s runner's actions are outside of any real human experience (I feel confident in saying that no-one has jumped down several stories to escape some form of giant robot), there is a core of real actions that the game simulates - running and jumping. However, the phenomenological experience of playing *Canabalt* is entirely distinct from the experience of real-world running and jumping. Running in *Canabalt* requires no input from the player at all. Jumping, similarly, requires no preparation, no tensing of the legs, and minimal energy - only enough to press down a key on a keyboard. The complex and often difficult process of sprinting and leaping over an obstacle is reduced to just a keypress.

Call of Duty 4: Modern Warfare (or simply *CoD4*) is significantly more complex in its controls than *Canabalt*, yet still fits into the same model of a reductionist correspondence. Here, I will focus on one particular element, the way in which reloading a weapon is represented. Firing, swapping, and reloading your weapons each have a single button dedicated to them on the controls. Pressing the fire button instantly fires a shot. Pressing the reload button instantly starts a reload animation, which completes itself in a quite short period of time (longer for some weapons than others - a pistol reloads very quickly, while a belt-fed machine gun is significantly slower). Pressing the swap button quickly puts away the current weapon and produces the second one. Each of these is significantly easier than the real-life equivalents (Smith, 2014). Reloading is particularly striking - the entire process of ejecting a magazine, inserting a new magazine, and, if necessary, loading a bullet into the gun's chamber is reduced and compressed down to pressing a single button. While *CoD4*'s overall controls may be more complex than *Canabalt*'s (involving controls for aim, fire, duck, reload, move, run, throw grenade, etc.), compressing the complex process of reloading a weapon like this is as striking a reduction as reducing jumping to a single button.

This analysis of *CoD4*'s reloading controls segues neatly into the simulation-control game that I will discuss, *Receiver*. A gun-simulator as much as a first-person shooter, in *Receiver* the player explores a series of apartments and industrial spaces populated by lethal drones, equipped with one of three different handguns and sometimes a flashlight. Each handgun (two kinds of semi-automatic guns and a revolver) has a distinct set of controls for reloading and preparing the

gun to fire. Taking the revolver as my example, there are different buttons to open the cylinder in which the bullets are held, empty the cylinder of its bullets, load new bullets in, close the cylinder, and pull back the hammer, at which point the gun is ready to fire. The other two weapons use magazines, and so have controls to load loose bullets into the magazine, to select a magazine from an inventory, and to pull back the slide on the gun, moving a bullet from the magazine to the chamber. This is a level of detail on the operation and control of guns that is diametrically opposed to *CoD4*'s single button press.

Receiver's focus on the detail of the operation of a gun acts also as a focus on the body, both the real body of the player and simulated (and entirely invisible) body of the character. Imposing on the player the need to shake out empty cartridges or to remember to release the safety makes the weapon, which is the primary focus of *Receiver*'s design in a way rarely seen in other games, feel like a physical object (Smith, 2014). There is an almost one-to-one phenomenological correspondence between the simulator action within *Receiver* and the simulated action of reloading a gun, at least in terms of complexity if not difficulty.

From these examples, it is easy to draw up a linear spectrum of phenomenological correspondence through games. At one end there is *Canabalt* and *Call of Duty 4*, defined by their reduction of complex actions to single presses. At the other end there is *Receiver*, defined by an attempt at complete accuracy of complexity - for every action in the simulation, there is an action on the controls. In between would be games with either a wider range of actions or a more complex approach to the control of those actions. This spectrum matches up closely to the spectrum of strongly to weakly serialized games described by Heilmann (2014, p. 38). Indeed, if one were to simply describe games as either reductionist or completist in their correspondence, then this paper would simply be a phenomenologically focussed version of Heilmann's seriality spectrum.

However, this would be an incomplete picture of the phenomenology of game controls. As well as the arcade-control model of a reductionist correspondence, and the simulation-control model of a completist correspondence, there is a third, relatively rare model, 'antisimulation'. Antisimulation as a control scheme represents an excessive correspondence between the phenomenologies of the simulated and the simulation. It can be seen in games such as *QWOP* (Bennett Foddy, 2008), *Surgeon Simulator 2013* (Bossa Studios, 2013), and *Octodad: Dadliest Catch* (Young Horses Inc., 2014), each of which are known for their awkwardness, difficulty, and strange control schemes.

By 'excessive correspondence' between the simulated and the simulation I mean that rather than reducing or matching the complexity of a particular action, an antisimulation game aims to increase the

complexity of the action that is performed. Several of the most popular antisimulation games in fact simulate very ordinary and simple actions, such as picking up objects in *SS2013*, walking around a room in *Octodad*, and running in a straight line without falling over in *QWOP*. Yet the way in which these normally simple actions are simulated leads to them being radically more difficult in simulation than in reality.

Though the obvious successes of antisimulation games such as *QWOP* and *SS2013* can make this argument on their own, it is worthwhile to state that antisimulation games are not bad games, and are not in some sense a failure. There is a strong assumption that a game with such a disruptive control scheme as *SS2013*'s would be a failed game, in part because of the presence of sincerely bad games with failed control schemes and in part because of assumptions about the value and importance of ease of use in games (Sommerseth, 2007, p. 767). However, antisimulation games show that this sort of assumption is ill-founded. A norm for games that assumes a 'relatively smooth process' of control and of learning the controls would be a valuable norm for arcade-control games - but by this point it should be clear that arcade-control games are only one of several possible options.

Referring back to Dreyfus's description of expertise in a tennis match, and the concept of expertise being given or not given through the control scheme, in an antisimulation game expertise is *actively withheld*. Even an expert player of *SS2013* appears clumsy, and *QWOP* transforms a common activity for most of us into a deeply difficult one (Bennett Foddy, 2010). Similarly, many antisimulation games work to undermine a basic element of our day-to-day experience, the ability to treat our own limbs as a singular system rather than a "mosaic of spatial values" (Merleau-Ponty, 2002, p. 112). Each leg of *QWOP*'s eponymous runner must be treated not only as a separate unit, but as two loosely linked units. In the case of *SS2013*, the surgeon, Nigel, only ever uses his left hand to perform all the actions - but the player must use both hands, one to control the arm and one to control the fingers. This example works as a standard for all of antisimulation. An antisimulation game is one where one simulated hand must be controlled by two real hands.

Antisimulators are a strange genre within games. Seemingly working against all the accepted norms of control schemes, accuracy, and power fantasy satisfaction, they undermine those norms by their success on different terms. Explaining their relationship to the more accepted control norms of arcade and simulation requires view all three approaches from a different perspective, that of phenomenological correspondence. This not only explains the way in which antisimulation controls act upon the player, but also the way in which arcade and simulator controls act. Seeing clearly the role of phenomenological correspondence relies on antisimulation, as the strange 'one hand/two hands' relationship between the game and the player puts the phenomenological relationship in to sharp relief.

Acknowledgements

Thanks and acknowledgements go to Alan Williamson, organiser of 'The Right Touch' topic at Blogs of the Round Table, where this paper originally began; to Peter Shafer at Ruminatron 5000; and to Jenni Goodchild and the audience at the 2014 Nine Worlds Academic Track.

References

Birnbaum, Ian (2014). Changing Arma 3's helicopters is Bohemia's first step toward a deeper simulation. *PC Gamer* [Online]. Retrieved 8 March 2015, from http://www.pcgamer.com/changing-arma-3s-helicopters-is-bohemias-first-step-toward-a-deeper-simulation/

Chandler, Daniel (1992). *The Phenomenology of Writing by Hand* [Online] Retrieved 5 May 2015, from http://visual-memory.co.uk/daniel/Documents/short/phenom.html

Digital Combat Simulator 'P-51D Mustang' [Online]. Retrieved 5 May 2015, from http://www.digitalcombatsimulator.com/en/shop/modules/dcs_p51d_mustang/?sphrase_id=1176782

Dreyfus, Hubert (1996). *The Current Relevance of Merleau-Ponty's Phenomenology of Embodiment* [Online]. Retrieved 27 February 2015, from http://ejap.louisiana.edu/ejap/1996.spring/dreyfus.1996.spring.html

Foddy, Bennet (2010). *QWOP* [Online]. Retrieved 5 May 2015, from http://www.foddy.net/2010/10/qwop/

Heilmann, Till A. (2014). "Tap tap, flap flap" Ludic Seriality, Digitality, and the Finger. *Eludamos* 8(1)

Live for Speed 'What is LfS' [Online]. Retrieved 15 February 2015, from https://www.lfs.net/whatislfs

Merleau-Ponty, Maurice (2002). *Phenomenology of Perception*

Richardson, Ingrid (2011). The hybrid ontology of mobile gaming. *Convergence: The International Journal of Research into New Media Technologies* 17(4)

Smith, Edward (2014). How to Use a Gun: Metro versus Modern Warfare. *Words That Won't Sell* [Online]. Retrieved 27 February 2015, from http://www.wordsthatwontsell.com/2014/09/how-to-use-gun-metro-versus-modern.html

Sommerseth, Hanna (2007). *"Gamic Realism": Player, Perception and Action in Video Game Play* [Online]. Retrieved 26 February 2015, from http://www.digra.org/digital-library/publications/gamic-realism-player-perception-and-action-in-video-game-play/

Games Cited

Aerosoft (2011) *OMSI*. Aerosoft (Windows)

Bennett Foddy (2008) *QWOP*. Bennett Foddy (Flash, iOS)

Bossa Studios (2013) *Surgeon Simulator 2013*. (Android, iOS, Playstation 4, GNU/Linux, Windows, Mac OS)

Infinity Ward (2007) *Call of Duty 4: Modern Warfare*. Activision (XBox 360, Playstation 3, Windows, Mac OS X, Wii)

Microsoft Game Studios (2006) *Flight Simulator X*. Microsoft Games Studios (Windows)

Semi Secret Software (2009) *Canabalt*. Semi Secret Software (Flash, Apple iOS, Android, PlayStation 3, PlayStation Portable, PlayStation Vita, Commodore 64)

Wolfire Games (2012) *Reciever*. Wolfire (Windows, GNU/Linux)

Young Horses Inc (2014) *Octodad: Dadliest Catch*. (Playstation 4, Playstation Vita, GNU/Linux, Windows, Mac OS)