# The design-by-adaptation approach to universal access: learning from videogame technology

IFAN D H SHEPHERD [A] • IESTYN D BLEASDALE-SHEPHERD [B]

*a) Department of Marketing & Enterprise*
*Middlesex University*
*The Burroughs*
*London, NW4 4BT*
*UK*

email: I.Shepherd@mdx.ac.uk

*b) Valve Software*
*10900 NE 4th St., Suite 500*
*Bellevue, WA 98004*

USA

email: Iestynne@gmail.com

Category: long paper

**Abstract:** This paper proposes an alternative approach to the design of universally accessible interfaces to that provided by formal design frameworks applied ab initio to the development of new software. This approach, *design-by-adaptation*, involves the transfer of interface technology and/or design principles from one application domain to another, in situations where the recipient domain is similar to the host domain in terms of modelled systems, tasks and users. Using the example of interaction in 3D virtual environments, the paper explores how principles underlying the design of videogame interfaces may be applied to a broad family of visualization and analysis software which handles geographical data (virtual geographic environments, or VGEs). One of the motivations behind the current study is that VGE technology lags some way behind videogame technology in the modelling of 3D environments, and has a less-developed track record in providing the variety of interaction methods needed to undertake varied tasks in 3D virtual worlds by users with varied levels of experience. The current analysis extracted a set of interaction principles from videogames which were used to devise a set of 3D task interfaces that have been implemented in a prototype VGE for formal evaluation.

***Keywords:*** *User interfaces, transfer, adaptation, videogames, virtual geographical environments, GIS.*

# 1.0 Introduction

> "*Look at games if you want to see a large number of user interface ideas, elegant and hideous.*" [16, p.308]

This paper is based on the premise that there may be significant benefits to be obtained in terms of universal access by transferring interface technology, or the principles on which such technology is based, from one application domain to another. In contrast to the more usual route of designing accessible interfaces specifically for target domains, through the use of formal design methods and frameworks, this paper illustrates an alternative and potentially highly productive approach to interface development, involving the adaptation of suitable interfaces from other domains [29]. This is referred to as the design-by-adaptation approach. Although this paper explores specific host and recipient domains (respectively, videogames and virtual geographical environments, or VGEs), the potential for such transfer extends well beyond these particular examples (see, for example, [84]). While it is not suggested that videogame interfaces represent a panacea for universal access, whether for VGEs or other software, it is contended that the transfer of widely used interface elements from such software is a potentially effective means of extending good practice beyond its original context. As dynamic 3D worlds become increasingly widespread in consumer and business software, so will the need for interfaces that ensure universal access to such worlds. Videogames provide a fertile source of 3D interface ideas, and design-by-adaptation offers a means for capitalising on such ideas.

Universal Access (UA) requires that all individuals experience equitable access to, and effective interaction with, information technologies [32]. This study does not explicitly consider specific users or user groups [65], nor does it examine specific user needs, disabilities [33, 52], or the role of user experience [10]. Nor does it discuss ease of learning, which is an important dimension of universal access that affects both the uptake and subsequent usage of information technologies, although it should be noted that many videogames adopt a learning-by-doing paradigm of learning, which is highly effective for users who have little propensity for reading instructions or time for undertaking formal learning in order to use information technologies. Rather, this study approaches universal access from the perspective of a single application domain whose interfaces have been designed to be usable by diverse users with multiple needs.

This paper is a continuation of previous studies by the authors into 3D data visualization [71], the convergence of videogame and VGE technology [73], and the transfer of learning [69, 72]. It was also triggered in a more specific way by the frustration of the older author (a VGE expert) at his continuing inability to master to a sufficiently high level the free-flying 3D interface controls available in some modern videogames, for which he envisaged potential roles in VGE software. This led to a joint search with the younger author (a videogame software engineer) for alternative interaction mechanisms among videogames which would enable flexible spatial manoeuvring in VGEs. It was intended that these mechanisms would be built into test-bed 3D data visualization software developed by the authors [70, 71]. However, lest it be thought that this paper represents a search for a solution to a particular individual's personal HCI problems, it should be noted that there is considerable research evidence which demonstrates that navigating in virtual 3D worlds can be extremely difficult and frustrating, even for relatively simple tasks [66]. Where highly complex 3D worlds are involved, the potential for users becoming disoriented or confused increases considerably [12, 31].

The design-by-adaptation approach does not imply that rigorous interface design and evaluation is ignored or bypassed. Rather, it involves the co-opting of already evaluated interfaces for related purposes, in a process with the following key steps: evaluation of domain affinity (in terms of software models, tasks and users); identification of potentially transferable interface controls and/or principles; adaptation of controls and/or principles for the recipient domain; software prototyping; and evaluation. This approach raises two important questions. First, what evidence is available to indicate the effectiveness of interfaces and/or interface design principles in the host domain? And second, is it possible to determine which interfaces and/or interface design principles will bring most benefit to a particular recipient domain? These questions are clearly intertwined, because it is only by comparing donor and recipient domains that a proper answer can be provided for the first question. In the remainder of this paper, we focus largely on the second of these questions, but we begin with an attempt to answer the first.

There are several reasons for suggesting the efficacy of 3D videogame interfaces. The first stems from the considerable testing to which such interfaces have been subjected, not only during the design process (e.g., by regular play testing with players and independent designers), but also by end users following purchase. A second line of evidence is to be found in the considerable volume of published research into the application of videogames in non-recreational domains. 'Serious games' are now a major focus for research and development in numerous fields: science [42, 63], education [83], business [55], military training [62], and environmental planning [25] purposes. If this research is to be taken at face value, then there appears to be a prima facie case for transferring videogame technology from its original domain. In contrast to most of these studies, however, the approach described here does not involve the use of videogame platforms for VGE applications. Rather, it considers more specific elements of videogame interface technology (including principles, techniques and mechanisms) that may be transferable to other application domains.

The paper is organised as follows. The first section undertakes a comparative analysis of the host and recipient domains (videogames and VGEs respectively) that are the focus of this study. The main body of the paper then critically evaluates interfaces developed for recent 3D videogames, and is organised around three themes that recur in studies of 3D interface design. Finally, based on principles arising from this evaluation, a number of 3D interfaces are proposed that might usefully be developed for VGEs, each of which is developed around a metaphor designed to assist its acceptance by novice, intermittent and discretionary users. Formal evaluation of these interfaces will be reported in a separate paper.

## 2.0 Comparison of videogames and virtual geographic environments

The first step in the design-by-adaptation process is the identification of relevant similarities between the source and target domain. It is well known that "the more the donor and recipient application have in common, the more likely that the technology will transfer successfully" [29, p.556]. Such comparison requires extensive knowledge of the two domains, and a careful analysis of those characteristics which might be conducive to or militate against interface transfer. The target domain considered in this paper consists of information systems that handle geospatial data. These typically display data in map form, and provide analytical tools for the solution of social and environmental problems. In common with videogames, which have a number of distinctive genres, geospatial data-handling software exhibit a number of fairly well developed niche types, including: desktop mapping (DTM) software, geographical information systems (GIS), remote sensing (RS) software, digital earths (DEs), and GeoVisualization (GeoVis) software. For convenience, these are collectively referred to in this paper as virtual geographical environments (VGEs). There is considerable overlap between these categories, with market-leading GIS software subsuming a great deal of the functionality offered in the other categories. There are also (largely experimental) crossovers between VGEs and virtual reality (VR) systems [26]. Because geospatial data are increasingly displayed as three-dimensional world views in VGE software, effective 3D interaction is becoming increasingly essential if users are to operate effectively in what for many will be challenging visual environments. In the context of this paper, it is taken as axiomatic that 3D virtual worlds are usable and productive to the extent that they provide effective user interfaces. Moreover, such interfaces will only serve the needs of the broad community of users if they are designed to ensure universal access. Because most VGE software in the past three decades has focused mainly on 2D data and 2D displays, the geoscientific community stands to benefit by looking elsewhere for ideas and principles with which to design effective interfaces for its emerging 3D environments.

For their part, many modern videogames share with VGEs a concern for representing geospatial data, either for real or imagined environments. There are, of course, some kinds of videogame (e.g., puzzle games, music games, and many children's games) which have little recognisable geographical content, and many sports games (e.g., tennis, soccer, snooker, American football, and Olympic events) which are played in venues with little intrinsic geographical interest. However, other games (e.g., golf and car racing games) are of considerable interest to the VGE community, because of their often fastidious and innovative approaches to micro-terrain modelling and townscape representation. In this paper, the focus is on videogames which have recognisable geographical content. These include: action games (and especially first-person and third-person shooters), role-playing and adventure games, several kinds of simulation game (including driving and racing games), and strategy games which involve the evolution of settlements and civilisations.

In addition to their representation of geographical environments, videogames and VGEs have several other features in common. For example, the tasks which users (whether spatial analysts or game players) undertake with such software have many similarities. At a high level of abstraction, and ignoring for the moment the self-evident difference between the entertainment purpose of most videogames and the scientific and policy support roles of most VGEs, both types of software are essentially problem-solving support systems operating in a spatial context. Tasks range in scope from the entire problem which the player or analyst has to solve, down to individual localized actions that need to be undertaken in moving towards completion of a global task. At a lower level of abstraction, several spatial tasks are common to both domains, including: exploration, navigation, searching, analysing, targeting, wayfinding, and the modification of world objects. The relatively informal classification of 3D interfaces by Bowman [8] and Bowman *et al*. [9], which distinguish three broad types of task that users commonly undertake in virtual environments (travel, selection and manipulation), applies to both videogames and VGEs, though the manipulation category is perhaps less common in current VGEs, and more prevalent in VR systems designed for object assembly. At an even more detailed level of analysis, Darken and Sibert [18] suggest that wayfinding behaviour, which is similar in scope to Bowman's navigation, can be subdivided into three largely mutually exclusive categories: naïve search, primed search, and exploration (essentially spatial comprehension). Over a decade ago, Treinish [79] underlined the importance of linking user-interaction methods to tasks in 3D visualizations. Task and interface taxonomies such as those described above provide a useful starting point in identifying common tasks undertaken with VGs and VGEs and linking these to interface designs which contribute to the effective completion of those tasks.

As a corrective for this picture of broadly aligned technologies, and for the implied potential for transferring technologies from one to the other, it is important to recognise that these application domains exhibit several significant differences. Many of these have been articulated by Pagulayan *et al.* [58] in a comparison of videogames and productivity software (a category which includes VGEs). To their ten points of difference (summarised in Table 1) may be added several others, including: the number of purchasers and users (hundreds of millions compared with tens of thousands); the age profile of users (increasingly wide in the case of videogames, but early-to-mid career in VGEs); the serial purchase of multiple titles by gamers compared with the purchase of perhaps a single VGE by organisations involved in spatial analysis and decision making; the frequent playing of games by many gamers compared with the infrequent use of VGEs by spatial analysts; the competitive research and development by thousands of games software developers compared with the far smaller number of geoscientific software developers; and the fact that few VGEs are populated by characters of any sort, whether they be heroes, monsters, zombies or skateboarders. It is also worth noting that while games software inhabit all points of Joiner's [35] continuum, from interactivity to storytelling, VGEs are mainly found at the interactivity end of the spectrum, though narrative has been introduced into a few dynamic cartographic and 3D Web environments by means of guided tours and other scripted devices [56, 15].

**Table 1**

Two further differences between videogames and VGEs are identified by Bolt and Tulathimutte [7] in their report of the usability testing of the *Spore* videogame. First, they suggest that the raison d'être of videogames is very different from that of more functional software (including VGEs), and thus their interfaces are more likely to be judged by whether they contribute to fun, engagement and immersion rather than to usability and efficiency. Secondly, they suggest that in many videogames, the player achieves a sense of satisfaction and control from the interface itself, and in such cases "interaction is both a means and an end". The inverse point is made by Manninen [44, 45], who suggests that videogame interfaces can sometimes be compensated for by compelling content, fast pace and fun. VGE software, in contrast, is unable to use such elements as a means of distracting the user from possible interface defects. These differences inevitably colour the interpretation of the evidence adduced from the exploratory research presented here.

Despite these significant differences, this paper takes the view that the similarities between these application domains are compelling enough to suggest that VGE developers can benefit from adapting some of the considerable range of approaches taken by the videogame industry in constructing effective interfaces for 3D virtual environments. To date, however, there appears to have been minimal borrowing from videogames by VGE development communities. This paper therefore attempts to fill significant gaps in both the research literature and software design practice.

## 3.0 Lessons provided by videogame interfaces

This section considers three important themes (degrees of freedom, viewpoints, and metaphors) which emerged from a study of videogame interfaces as being significant in contributing towards their usefulness in VGEs. Although these are by no means exhaustive (other significant themes include: multimodal interaction and multi-sensory feedback; affordances; behaviour amplification; and multi-player interaction), they nevertheless provide initial insights into the principles required for the design of effective 3D interfaces for VGEs. For each theme, principles of effective design are proposed and, following a discussion of the third theme, interface mechanisms based on metaphors which have been prototyped in 3D data visualization test-bed software are described.

The analysis conducted by the authors consisted of a form of content analysis of recent videogames in order to identify effective interface design principles and mechanisms. This was partly an inductive process, grounded in detailed game play. However, the authors' knowledge of VGEs was used throughout the process to guide the identification and subsequent selection of high-potential transfer candidates. Additional information was provided by five experienced videogame designers at Valve Software, who were interviewed for a parallel research project into the design of videogame affordances [74], some of whose ideas are relevant to the current analysis.

As previously noted, the unit of analysis in this study is the specific or ‘local’ interface mechanism. This type of mechanism, referred to here as the *task interface,* in order to distinguish it from the user interface as a whole, constitutes the particular interactional tool used to perform a specific task. The task interface may be considered as the mediating link between human input and computer output, or as a mapping between controller usage and task-related behaviour within the virtual environment. Each task interface occurs in a specific region of virtual space-time during gameplay. However, it should be noted that while there are typically a large number of task interfaces in any given videogame or VGE, many are used repeatedly, while only a few are devoted to specific tasks. Videogame designers interviewed for this study refer to two types of task interface: ‘broad’ or ‘general-purpose’ on the one hand, and ‘narrow’ or ‘custom’ on the other, with the first being pressed into service in as many contexts as possible.

## 3.1 Degrees of freedom

Interface devices represent the physical medium through which most software users accomplish tasks. Although some tangible interfaces establish a direct correspondence between what the user does with a device and what happens in the virtual environment [19], and some devices (e.g., the cubic mouse [22]) have been built to work with highly specialised virtual environments, it is rare for a particular device to have only one interface role -- i.e., for the user of a device to be able to accomplish a unique effect while using it. In order for devices to be used with a variety of software and for a variety of tasks, device operations must be capable of being linked to user operations in some flexible way. In a games context, Pagulayan *et al*. [58, p.892] argue that “the way that functions are mapped onto available input devices can determine the success or failure of a game”.

One way of evaluating the contributions that particular devices make is to consider their broader, non-hardware characteristics, such as the design spaces they inhabit (e.g., [13]). The approach taken here considers how devices enable users to control their software effectively to accomplish required tasks. The analysis is therefore directed more towards the set of controls included on various devices (e.g., buttons, wheels, paddles, joysticks) and, more specifically, on the degrees of freedom that devices provide and how these are associated with the degrees of freedom required by particular software and user tasks. The analysis will focus on task interfaces within videogames which provide ideas that may prove useful in creating more effective interfaces for VGEs.

### *3.1.1 The relevance of degrees of freedom*

One way of thinking about interface devices and their utility is to consider the degrees of freedom (DOF) that they offer users for interacting with 3D software. Indeed, interface devices are often described in terms of the DOF they provide. Thus, for example, versions of the humble mouse are available with 3, 4, 5 and 6 DOF, while more complex devices, such as hand-fitting gloves and body suits, commonly have over 20 DOF. The devices available to most videogame and VGE users have six standard degrees of freedom. These include three DOF for viewpoint or player *position* (relative to the x,y,z coordinates of the 3D scene), and three DOF referring to the *orientation* of the viewpoint or player at a given location (these are conveniently defined using the aeronautical terminology of pitch, roll and yaw).

In design terms, it is often more productive to think in terms of the DOF made available to users of particular software by software mappings, rather than the hardware device being manipulated, which in many cases is a given. There are several reasons for this. First, the software might require fewer or more DOF than the user's device might be capable of. Second, the number of DOF a user needs to manage may change during the course of using the software, for example as they move from one task to another. And third, the mapping of particular DOF to device controls is likely to change during a user session. In videogames, it is also often the case that different players of the same game may use different input devices; this is most often true for games which are available on multiple hardware platforms. Table 2 indicates how a selection of contemporary 3D videogame interfaces runs the gamut of the six standard DOF.

| **Table 2** |
|---|

### *3.1.2 Complex DOF mapping*

In reality, classifying interface devices in terms of DOF is not usually as simple as the examples in Table 2 might suggest. For example, the DOF presented to the player by an input device may be mapped in a considerable variety of ways to one of many DOF of the state of the game's virtual environment. It should also be noted that, even if a virtual DOF represents a traditional viewpoint DOF (position/orientation), input DOF may be mapped onto the *velocity* or *acceleration* of any of the positional and orientational DOF. Beyond this, the physical DOF might also map to any number of other DOF of the state of the virtual environment. Common states include: the position/orientation of the player's *avatar* (which, as will be discussed later, may not be at the same position/orientation as the viewpoint), and the direction/strength of a force in the environment (e.g., wind or gravity).

It may therefore be useful to distinguish between *physical DOF* and *virtual DOF*. Physical DOF refer to the number of independent controls available on the interface device, and virtual DOF refer to the number of independent controls instantiated in the virtual environment, in terms of viewpoint position and orientation. It is the application software that determines how physical DOF are mapped onto virtual DOF. The control scheme used in the console game *Pikmin*, for example, effectively maps 2 physical DOF to 4 virtual DOF. The way this is implemented is that the controller joystick usually moves a target point but, when fully tilted, it moves the player's avatar as well as the target point, so the player can position them both with the same joystick. The manner in which intermediary logic (such as simple inertial delay) can increase the complexity of a simple physical DOF is very interesting. A good physical example would be a player swinging a weight on a rope by holding the rope at the top and moving their hand from side to side. It might seem at first sight that this 'interface' offers a mere 1 DOF, but in fact the state of the system is not a simple mapping from the state of the player's hand to the state of the weight - the mapping is in fact a function of input state *over time*. This is essentially a very large vector[1].Vehicle control in the game *Halo* uses this concept; one joystick rotates both the viewpoint and the vehicle, but the response of the vehicle lags significantly behind that of the viewpoint. This interface subtlety allows the player to control both the viewpoint and the vehicle with a single joystick, with a useful degree of independence.

### 3.1.3 Allocation of control/device/interface DOF

Classifying 3D virtual world software according to the degrees of freedom that their interfaces offer users may appear to be a rather esoteric pastime. However, as Masliah and Milgram [48] have shown, it is a requirement of effective interface design that designers understand how users prefer to allocate their controls across the degrees of freedom available

---

[1] Considering the hand's position, sampled over the last 5 seconds at a rate of 10Hz, to be sufficient to determine the current state of the system, then 50 input values need to be considered, and hence 50 DOF.

to them. More specifically, they point to a lack of empirical information about whether, in the context of complex, high DOF input devices, the optimal distribution of DOF should be across one or two hands. The outcome of their experiments confirms previous psychological research which indicates a preference among individuals to allocate control to separate rotational and translational groups, and to switch between these groups when undertaking docking tasks. These experiments also indicate significant differences in behaviour between novices and expert users in the presence of 6 DOF devices, with novices being more likely to reduce task complexity by controlling only a subset of the available DOF at any one time, and experts being more likely to allocate control across all DOF. These authors also suggest that the equal allocation of control among all DOF available in input devices is unlikely to provide an optimal solution in all situations (when compared to unequal control), as it will depend significantly on the nature of the task, the environment, and the users.

### 3.1.4 Simplification of allocation of control/device/interface DOF

Several interim conclusions may be drawn from these observations. The first is that restricted freedom of interaction is commonplace in videogames, even when the available input-output devices permit higher degrees of freedom. A key characteristic of most modern videogames is that, despite the apparently complex 3D environments in which players exercise their problem-solving skills, very few provide complete freedom of movement. Indeed, where 6 DOF are provided, as in the first-person game *Descent*, it has usually proved difficult to master, even using a high-end joystick, and sales have tended to be lower than expected. In most first-person and third-person games, the player’s movement is usually constrained within the 3D gameplay world to 3 or 4 DOF. In most modern games, for example, players move forward and back, and strafe (or pan) left and right, but their feet or vehicles stay largely on the ground (except for the occasional leap). As for orientation, this too is usually restricted to pitch and yaw, with roll rarely being required. This means that most games can be played using a combination of standard mouse and keyboard controls. Thus, despite the apparent 3D nature of many videogame worlds, the gameplay itself is typically constrained to a surface, and this suggests a means of limiting the interface complexity which needs to be mastered by 3D VGE users.

In some cases, videogame interfaces reveal considerable ingenuity on the part of the designer in reducing the DOF under player control. As previously described, in the console game *Pikmin*, a single joystick is used to control the movement of both the game’s main character and the chosen target, with the angle of the joystick determining whether one or both of these move in the direction indicated. Such domain-specific and task-related interface design is commonplace in 3D games. In the *Dogfight* aerial combat game, full control of the aircraft is possible using the mouse alone, because of two inherent movement constraints: the aeroplane can only move forwards and, under normal circumstances, cannot remain motionless. Similar restrictions on the control of viewpoint movement are found in the console game *Katamari Damacy*, in which two joysticks are used to move the game object around the surface of the 3D world, using a ‘tank track’ form of control. (The complexity of controlling avatar motion across planetoid surfaces is similarly reduced in *Super Mario Galaxy*.) Despite this game’s addictive gameplay, the avatar has 3 DOF, but the interface surprisingly exposes only 2 degrees of freedom to the player via the controller. This game also adopts a navigation technique common to many first-person and third-person games (e.g., *Jak and Daxter* and *Half-Life*), which reduces the complexity and also increases the naturalness of user interaction. This involves viewpoint-relative motion, in which the user moves in the direction of current viewpoint yaw. It should be noted that while few first-person games provide absolute motion control, software used for map editing by spatial analysts is routinely used in digitiser mode.

In some real-time strategy games, such as *Sim City*, it is possible to reduce the degrees of freedom still further by fixing the viewpoint position at a constant angle with respect to the isometric grid play area, so that the mouse alone is needed to zoom and pan with respect to the play surface. The quasi-3D weather map broadcast daily on BBC TV imposes even greater constraints, in that the viewer is always looking at the map from the same direction and height, and at the same vertical viewing angle, resulting in essentially 2 DOF interaction by the weathercaster. This serves to minimise potential viewer confusion, especially since they are not in control of movement across the map.

### 3.1.5 DOF conclusions

The main conclusion to be drawn from this analysis of videogame interfaces is that there is much to be gained by simplifying the navigational facilities made available to users of 3D VGE software, and by providing a *constrained navigation* approach. In this vein, Hanson and Wernert [28] propose the use of only 2 DOF for moving across 3D terrains. Evidence from more complex 3D VGEs, such as those developed in the oil industry, suggests the probity of this strategy, because providing complete freedom “requires considerable ability from the user, often resulting in spatial disorientation and object collision” ([12],  p.176). There is a growing body of research evidence from simplified lab-based tasks (e.g. [78, 27]) that provides guidance to designers on this issue, but further research into the use of various interfaces for complex decision-making activities in 3D worlds is urgently required.

A further conclusion is that there is often a disparity between the degrees of freedom afforded by the input hardware available and the degrees of freedom instantiated by the game software. Indeed, in most cases, through the intermediation of device drivers and interface software, there is an indirect relationship between hardware and interaction facilities. Just as specific characters are mapped onto specific keyboard keys, so certain degrees of freedom are mappable onto varied input hardware. In designing interfaces, it is therefore important to focus on the complete task interface package: the physical controller usage, its mapping onto interface DOF, and the mapping from there onto virtual environment state variable DOF.

The following interface design principles are suggested on the basis of the foregoing discussion:

Principle 1: Minimise task interface complexity -- for example, by adopting constrained interface models that reduce device DOF.

Principle 2: Adopt alternative ways of mapping device DOF to software DOF for different users and tasks, thereby creating a family of task interfaces.

## 3.2 Viewpoints

An important aspect of any 3D visualization is the way in which the viewer (i.e., a game player or data analyst) is positioned with respect to the displayed scene. This section discusses several significant aspects of user positioning and extracts principles relevant to VGE design.

### *3.2.1 Centricity*

The location from which the viewer observes the scene is known as the viewpoint or frame of reference [82]. In videogames, the camera is positioned at the viewpoint, and the choice of viewpoint therefore largely determines whether the game provides a first-person or third-person experience. In first-person games, the viewpoint is attached to the player, and serves to contribute a greater sense of immediacy. In third-person games, in contrast, the viewpoint is usually attached to a character with an independent identity whose actions are under the player’s control [16]. Few games adopt a second-person viewpoint -- i.e., where the viewpoint is attached to someone other than the player. In some studies of virtual environments (e.g., [53, 54]), the positioning of the viewer with respect to the scene is referred to as the centricity of the display. Two types of centricity are generally recognised: the egocentric viewpoint (typical of first-person games), and the exocentric viewpoint (typical of third-person games). Kruger *et al.* [38] refer to these viewpoints as the 'inside-looking-out' and 'outside-looking-in' approaches respectively, while Ware and Osborne [81] refer to them respectively as the 'eyeball-in-hand' and 'world-in-hand' metaphors. In VR systems designed for the close manipulation of objects, the exocentric viewpoint is also sometimes referred to as the object-centric viewpoint [50].

The efficacy and enjoyment of the interaction experience, in VGEs as well as videogames, is in no small way related to the choice of viewpoint. The egocentric viewpoint is generally the more immersive, in that the viewer not only has a sense of being *at* the viewpoint, but is also in the thick of the action. In videogame terms, the first-person camera is part of the scene; through it, the player is *involved*. The head-coupled devices used in VR systems can considerably increase this sense of immersion or involvement. (A discussion of the measurement of immersion in games is provided in [34].) The exocentric viewpoint, in contrast, usually lacks the same degree of immersion, but compensates by providing a broader or synoptic field of view of the scene. In videogame terms, the third-person camera is not part of the scene, but adopts what is commonly referred to as the 'god' perspective. The player tends to be perceptually, cognitively and affectively more detached.

The design of many contemporary videogames suggests the value of making a qualitative distinction within the exocentric category, to take into account the embeddedness of the viewpoint within the scene. While the exocentric viewpoint in many military or battlefield simulations tends to place the viewpoint in a fairly detached, lofty position to enable commanders to benefit from a wide field of view, in many third-person and vehicle driving games the viewpoint is typically positioned within the scene itself (e.g., behind and/or a little above the rear of the vehicle), and is therefore far closer to the player (in the form of the driver) and to the game action. The distance of the viewpoint from the player clearly has a significant bearing on the experience of immersion during gameplay, and on the degree of control players are able to exert on the vehicle they are driving. Videogames therefore suggest that a relatively high degree of involvement can occur between the egocentric and exocentric positions. An example is provided in games such as *Resident Evil 4*, where immersion is key. In this particular game, the third-person camera gets close to the player, almost sitting on their shoulder like the proverbial pirate's parrot. This is the 'over-the-shoulder' view in videogame parlance.

Viewpoint choice has major implications for the design of VGEs. Detailed studies have been undertaken in a military context (e.g., [4, 49, 77]) which reveal that the egocentric and exocentric viewpoints are suited to different tasks and benefit users with different levels of experience. Since some of the tasks undertaken in these military studies (e.g., judgements of relative position, mobility assessment and line-of-sight visibility) are similar to those used in VGEs, it may be useful for VGE software to enable users to switch as needed between the two major types of viewpoint as is possible in several videogames. In the first-person *Brothers in Arms* and *Ghost Recon*, for example, the player can momentarily switch to a third-person camera to provide a tactical overview when the combatant has taken cover. The roleplay game (RPG) *The Elder Scrolls III: Morrowmind* also offers first- and third-person viewpoints, with the latter being useful when the player's avatar wishes to display its winnings. In some games, the viewpoint is switched automatically, either to reveal important events (as in *Civilization: Revolution*), to taunt opponents with amusing character animations or for replay purposes (as in *Team Fortress 2*), or to reveal the player's avatar when it is injured (as in *Left4Dead*).

The switching between viewpoints is just one of several interface facilities that contribute towards universal access. In most 3D virtual world software, there is generally too much variety (in datasets, environments, problems, goals and tasks) to expect that a 'one size fits all' interface will be universally effective for users. The ability to switch effortlessly between task interfaces at any relevant time during an interactive session may therefore be desirable, and could be implemented in several ways. One is to adopt the modal approach, in which the user decides what will be changed in the interface, and when. This approach is similar in spirit to migratory interfaces [3, 59] and user-adapted interfaces [64], and is represented in several videogames by user-controlled viewpoint switching. A second approach consists of automatic interface switching by the software in response to detectable changes in the player's task-related behaviour. This AI approach is also common in videogames, especially where the software determines when and how player-controlled avatars should make transitions between walking, running and climbing.

### *3.2.2 Control coupling*

In a large proportion of 3D videogames, the viewpoint is not controlled directly by the user. Rather, and to some variable degree, the viewpoint is placed and moved automatically by the software, in reaction to the user's control of a focus point within the environment. This focus point is often -- but not necessarily -- the player's in-game avatar. God games, for example, may not have a precise or standard focal point, and a number of games (e.g., *Zelda*) change the focal point temporarily to a targeted enemy, while point-and-click adventure games (e.g., *Myst* or *The Crimson Room*) focus the camera on objects in the environment rather than on a player avatar. Despite these examples, however, it is not common for videogames to focus on something other than the player avatar. One axis along which videogame navigation interfaces may thus be classified is the degree of (de)coupling between the viewpoint and the focus point. As Table 3 illustrates, *viewpoint-focus coupling* in 3D games runs the gamut of possibilities between complete coupling at one end of the spectrum to complete decoupling at the other. As one moves from the former to the latter end of the spectrum, the player has to manage an increasing number of DOF. The entries in Table 3 suggest that control DOF are independent of the amount of coupling. With good reason, there are probably no games in which the player has to manage both the position/orientation of the camera and the avatar explicitly but independently. Nor are there any games in which the position and orientation of the avatar is handled by the player while the position and orientation of the decoupled target(s) is managed by the software.

**Table 3**

As previously mentioned in connection with driving games, centricity cannot be entirely detached from the issue of user control. One of the more apparent differences between geo-scientific and videogame virtual worlds is that while in the former the user almost universally has control of only one entity (typically the viewpoint), in a large proportion of videogames the user has control of two (and sometimes more) entities, typically the viewpoint and the player’s avatar. While the presence of an avatar representing the player (and/or other characters) within the scene is a common feature of many videogames, it is conspicuously absent from most VGEs, despite its role in indicating the principal point of interest or focal point within the game. This is especially true of third-person games, where the viewpoint is always located at some point other than the player. However, the presence of an avatar representing the player does not necessarily label a game as third-person, because this is more to do with whether the camera represents the avatar's eyes. In first-person games, for example, the avatar's weapon is often visible, and in *Mirror's Edge*, great play was made in the game's original publicity of the way in which the avatar's hands and feet are visible as she teeters on the edge of high buildings.

The use of avatars in games provides several advantages for players: it provides visual feedback on their movements (e.g., when they are jumping, fighting, skateboarding, skiing, swimming or driving); it allows them to see what is going on in the vicinity of the avatar (e.g., enemies sneaking up behind them); and it provides a strategic view of the local environment (e.g., to plan paths and execute jumps). The state or behaviour of the avatar indicates the actions being performed and helps track success. For example, in a skateboarding game (such as *Skate*) the player sprawling face-down on the floor is a good indication of failure, whereas their landing a jump neatly and punching the air is an indication of success. This is a key feedback mechanism found in most games. While such animations might not readily find a place in more 'serious' software, a significant role for a player avatar in VGE software might be to indicate the observer's location in a scene (something that Microsoft's *Digital Earth* attempts to provide), and to indicate what might be visible from a particular location (e.g., by displaying an intervisibility sector on an exocentric view with respect to an avatar's location).

In addition to controlling their own avatar, game players may also control a target object, such as another character's avatar. In videogame environments, this means that the user may need more than the usual maximum of 6 DOF, since they will need several DOF per controllable entity. This control may be achieved through dedicated buttons on a controller, or by mode switching. It should be noted, however, that mode switching differs from context-sensitive switching in that the former is user selected while the latter is software enacted. This introduces a further significant ingredient to the current analysis: the relationship between two sets of object controls. Here, a second form of coupling may be defined, *object control coupling*, which is the technique used by the player to control more than one object (typically their own avatar and that of another character, but perhaps their own avatar and a game world object) using a single set of game controls. In general, however, not many games require the player to control multiple things at the same time (with the exception of the idiosyncratic *Pikmin*). Usually, the avatar is frozen while the player controls another object, or else the player switches between multiple avatars. One common tactic is for the player to issue simple commands to other characters (e.g., "go here" while pointing at a target location, or "wait here", or "follow me") as in *Half-Life 2*. This would seem to suggest that although some forms of coupling are definable in principle (for both object control coupling and viewpoint-focus coupling), they do not always find their way into the repertoire of videogame control mechanisms, usually for reasons related to ease of use.

One further complexity needs to be added to this discussion of viewpoints and user control, which concerns the effect of the display surface through which most user interaction occurs. The visual world/scene depicted in a game or VGE is typically represented on screen within one or more windows, which may be displayed on one or more screens. The design of the contents of these windows has a significant bearing on the usability and enjoyment of the game or VGE. Clearly, where the scene being visualized is represented in more than a single window, care must be taken to ensure congruence and synergy between them [60]. One of the advantages of using multiple windows and/or screens is that several viewpoints, both egocentric and exocentric, are viewable at all times. However, while this might improve overall scene navigability and the user's ability to position and orient themselves within a scene, it may also lead to loss of viewer immersion, divided attention and interaction delays due to switching between windows.

Multiple windows represent what might be called multiple focus interfaces, and the technique has been experimented with in films such as *Timecode* and *The Andromeda Strain*. However, although this has never been widely adopted in the cinema, perhaps for reasons that might be instructive to both game and VGE designers, it has found greater favour in CAD and 3D graphic design systems, where the operator is often afforded several complementary viewpoints of an object or scene. If multiple windows are present at all in games, they tend to appear in multi-player games (e.g., car driving/chase games), where they provide an individual visual focus for each player. Few (if any) games provide a multi-focus interface -- e.g., have windows which show different views inside and/or outside a vehicle. The main reason for this absence is that a single window helps to maintain as high a degree of player immersion as is possible in a desktop environment. The absence of on-screen menus and informational windows serves a similar immersion-inducing purpose. Clearly, in applications such as VGEs, in which user immersion is traditionally not a priority, multi-focus interfaces are more attractive, especially where individual windows either provide complementary views of the same scene (as in Plumlee and Ware's [60] coupled views), or complementary types of data visualization (as in the linked or coordinated views of [1, 24, 47]). This is one area in which videogame practices are unlikely to be transferable to VGEs in a modified form.

### 3.2.3 Navigational aids

Although viewpoint choice and coupling tools are important in enabling users to find their way through virtual environments, other facilities are often needed, especially in egocentric displays. Knowing where the users are located, as well as where they are facing or heading, is important for effective navigation in 3D virtual worlds. Indeed, positioning and self-orientation aids are equally essential to videogame players and VGE explorers. Some videogames (e.g., *Grand Theft Auto: San Andreas*, *Midnight Club: Los Angeles* and *Far Cry II*) include a circular mini-map which shows local street details or landscape features around the player's current location. However, surprisingly few 3D videogames provide spatial orientation or directional indicator aids, and thus provide relatively few lessons for VGE designers. One of the reasons for the absence of such aids lies in the design imperative of maintaining the illusion of immersion throughout the game. Indeed, for some games, uncertainty about the player's location and orientation may be part of the game's appeal. Another reason for the general absence of map-like orientation aids in videogames is that the player is expected to acquire detailed knowledge of the game environment, typically through rapid learning and repeated intensive play. One game designer [17, p.234] suggests that "when you want to encourage exploration, you want to make sure that maps are unnecessary." Little, too, may be learned in this regard from several centuries of 2D mapping, though several 3D adaptations of the 2D north arrow have been devised for websites (e.g., in the oblique map interface on the Map24 website), and floorplan indicators are widely used in digital museum guides. However, these are essentially planar devices which may be well suited to worlds offering constrained 3D navigation, but they are far less relevant to full 3D VGEs with higher degrees of freedom of movement and orientation (as discussed in section 3.1). The 3D 'floating north arrow' devised by Micro Images [51] to help orient those flying around digital elevation models is an example of poor adaptation of the 2D version, because it loses its directional capability when the symbol is viewed from a low angle. VGE designers may therefore have little to learn in this respect from designers of games worlds or indeed from conventional digital mapping software, in which the map is the view. Additional research specific to 3D geodata visualizations may therefore be needed to develop more effective aids for positioning and orientation.

The following interface design principles are suggested on the basis of the foregoing discussion:

Principle 3: Find the best mix of viewpoints for specific tasks, and allow users to switch between them.

Principle 4: Where appropriate, use control-viewpoint coupling to reduce interface complexity.

Principle 5: Introduce avatars to support specific navigation and search tasks.

## 3.3 Metaphors

Although it has been argued in a previous section that interface design can be improved by considering the degrees of freedom made available to users of particular games or VGEs, players and analysts alike are unlikely to be aware of this way of thinking about their task interfaces. When playing a game, users typically think (intuitively rather than analytically) in terms of affordances – i.e., what am I enabled to do with this interface; or what am I being, doing or imitating when using this interface? In asking these questions, players often consider how their interaction links to previous actions they have undertaken in real and/or virtual worlds. This indicates the importance of exploring the role of *metaphors* in 3D interface design.

### *3.3.1 How interface metaphors work*

Interface designers have frequently adopted metaphors to relate what appears on screen to the user's personal experience. Metaphors are effective to the extent that they enable the transfer of elements of user experience to the mental model being developed as they use a game. When a metaphor is used to refer to interactions undertaken by users at the interface, it brings some degree of familiarity to potentially unnatural operations, by associating them with the user's own real-world interaction or navigational experiences. Several metaphors, such as windows and dashboards, were devised in the early days of interactive computing, when many of the principles of metaphor design were expressed in the design of 2D graphical user interfaces [14, 20]. Interestingly, Mark [46] argues that most computer interfaces are based on spatial metaphors. In the context of modern VGEs, metaphor can be used in two broad ways: to represent the content and structure of virtual worlds, and to represent user interactions within those worlds. Although the focus here is on the second of these uses of metaphor, it should be noted that the two can rarely be entirely separated in practice. Thus, for example, the map metaphor widely used in VGE software to refer to the structure of representational space as maps is inextricably linked with the manipulation of map layers, and with the process of spatial overlay analysis.

Since the 1980s, the map metaphor has been almost universally adopted in GIS and desktop mapping software designed to handle spatial digital data. Even by the early 1990s, however, Kuhn [39, p.460] suggested that this particular metaphor was hindering recognition of "the tremendous potential lying in non-map-based forms of interaction with spatial phenomena". He went on to suggest [40] that user-level metaphors rather than implementation metaphors were needed to guide the design of effective interfaces. As 3D virtual worlds have increased in popularity, many user-oriented metaphors have emerged. A common approach adopted where non-spatial multidimensional data are being visualized has been to design virtual worlds that reflect natural scenes, landscapes or 3D environments such as museums, gardens, shopping mall, hotels or cities. Other metaphors, based on spatial processes rather than physical environments, have also been devised. Examples of metaphors devised to assist user interaction in 3D virtual environments include: the handheld virtual tool [2], the World in Hand or WIM [76], the magnetic attraction metaphor [37], the peephole metaphor [11], the improved virtual pointer [75], the K-Cube [57], and anthropomorphized objects [67]. At a broader level, metaphors such as (direct) manipulation, immersion and presence have also been widely adopted.

### *3.3.2 Interface metaphors in videogames*

Numerous metaphors have been adopted in videogames, not only for obviously spatial tasks such as navigation and using weapons, but also for selecting game options and acquiring gameplay information [61]. Of particular note is the composite metaphor of the combined radar and compass navigation object used in *X2 The Threat*, which suggests a more appropriate navigational tool for 3D VGEs than 3D adaptations of the conventional north arrow discussed earlier. Examples of other metaphors commonly used in contemporary videogames are summarised in Table 4. One of the key objectives that videogame designers try to achieve when choosing interface metaphors is the minimisation of the need for formal user training before being able to play a game. Well-chosen and well-implemented metaphors enable interface operations to be rapidly assimilated by users during the process of playing the game [61]. Because VGEs usually require considerable prior learning before being usable, the transfer of a 'learning through doing' design principle from videogames may require considerably more adaptation than other design principles, because it impacts on wider working cultures that typically involve formal education and training.

| **Table 4** |
|---|

It is readily apparent from Table 4 that many of the metaphors used in videogame virtual worlds relate to actions that humans undertake in the physical world, while remaining appropriate to the fiction of a given game. Also of note is the way in which the same metaphor may be implemented differently in different games, and may be associated with different operations in different games. The running metaphor, for example, is implemented differently in the three PC games *Half Life 2*, *Escape from Monkey Island* and *Alien Swarm: Infested*, and in the three console games *Jak & Daxter*, *Resident Evil*, and *Killer 7*.
This evidence would seem to suggest that the chief role of an interface metaphor is to facilitate acceptance by players of the general nature of an interaction mechanism, rather than to provide detailed clues as to how that mechanism works, or how a particular effect should be achieved. The tether metaphor in Table 4, for example, might tell the user how the display will behave as they adjust relevant interface controls, but it does not provide a recognisable model of how these controls should be adjusted. Metaphors which explicate how a mechanism works are found in very few games. (In the console game *Fight Night Round 3*, for example, the player swings the two joysticks up and around to imitate the throwing of punches.) However, with the appearance of more naturalistic controllers (e.g., the Nintendo Wii, and Microsoft's Project Natal interface, both of which interpret player gestures), and tangible interfaces (such as some augmented reality table-top displays), a closer link is being established between the interface metaphor and the actions required of the user to operate a certain mechanism and achieve a required outcome.

### *3.3.3 The challenge of designing interfaces with metaphors*

There are several potential problems in designing interfaces for 3D virtual worlds based on metaphors. Schneiderman and Maes [68], for example, suggest that poorly designed interface metaphors may create unrealistic user expectations, undermine system predictability, and reduce user control in certain circumstances. Franck [21] has also suggested that placing too much emphasis on the surface representations of interfaces (e.g., on icons, rather than on windows or direct manipulation) may lead to the underpinning metaphor being ignored or underused. For their part, Barr *et al.* [5, 6] suggest that one means of ensuring the successful use of a metaphor is to undertake a thorough analysis of metaphor entailments[2] as an early part of the interface design process. The authors draw particular attention to the importance of giving careful consideration to metaphoric entailments when choosing metaphors for user interfaces. This is especially necessary in cases where novel metaphors are being adopted, because their entailments are unlikely to be firmly established, and the interface is therefore unlikely to be able to draw on existing entailment knowledge.

### *3.3.4 Some interface metaphors for VGEs*

An important point to note about effective interface metaphors is that they must be carefully chosen to fit the content and purpose of a particular virtual world. This means that metaphors used successfully in one application domain may not necessarily be transferable to other domains without significant adaptation. Indeed, some effective interaction metaphors for 3D VGEs may need to be purpose-built for specific applications, though it may be possible to select from an existing palette of carefully crafted metaphors. With this in mind, several additional interface metaphors for VGE software are proposed (Table 5a), based on videogame design principles. These not only accord with the main principles of effective metaphor design outlined earlier in this section, but also adopt some of the principles discussed in previous sections, especially those relating to degrees of freedom and viewpoints. Because of the significance of travel tasks in VGEs, the seven proposed metaphors refer mainly to tasks which in a VGE context involve movement and inspection (i.e., visual exploration). A key feature of the proposed interfaces is that, except for the one adopting the free flight metaphor, they all involve some degree of constraint in the use of the 6 DOF nominally exposed by the interface devices available to most VGE users (typically mouse and keyboard on the PC, and one or more joysticks and buttons on a game controller).

| **Table 5a** |
| --- |

No claim is made for the novelty of these interface metaphors. Indeed, the free flight metaphor is available in many videogames, and the terrain following metaphor has been adopted by both Google Earth and Microsoft's Digital Earth [41]. In order to evaluate the effectiveness for VGEs of the interface metaphors listed in Table 5a, they have been

[2] Metaphoric entailments are the implications of the signifier -- i.e., the metaphor -- about the signified -- i.e., the interface.

implemented in the experimental 3D data visualization software created by the authors, and are currently being user tested.

Each of these interfaces requires the software to undertake some of the navigational work on behalf of the analyst. In the tracking interface, for example, the software not only takes care of the basic tracking of a moving object, but it also monitors the user's inputs so that manual adjustments may be made along the way (e.g., to the viewing angle or distance). In the guided missile or arrow interface, the software computes a suitable trajectory from the current user location to an intended new location, as well as the acceleration and deceleration parameters needed at the start and end of the path to avoid making the journey visually unsettling. The simplest path might be a straight-line or parabolic curve, the latter echoing the Google Earth facility for jumping from one location on the earth's surface to another. Other studies have gone further, introducing various forms of path-planning aids [43], guided navigation facilities [23] or fully automated movement [12]. Such interface tools would help users avoid intermediate objects along the path, in a manner demonstrated by astronauts flying through asteroid debris in such films as *Armageddon*, or skydivers in such games as *Ratchet & Clank*. The technique of making jump cuts from one location to another, pioneered in cinema, and more recently adopted by some videogames [16], might be less acceptable in a VGE context, where the analyst's traversal of the virtual environment is frequently accompanied by visual search and inspection. Returning to the discussion in a previous section, it could be argued that the last two interface metaphors involve zero degrees of freedom, because the user's movement is entirely under software control. However, in the authors' test-bed implementation, the user is able to override most of the viewpoint positions and orientations while moving along the path.

| **Table 5b** |
| --- |

Following Barr *et al.*'s [5, 6] precepts, care has been taken to ensure that none of the metaphors evoke too many entailments, that few of the entailments implicit in the chosen metaphors have been ignored, and that there are (hopefully) no examples of mixed metaphors. However, as Table 5b reveals, several of the entailments evoked by the authors' proposed metaphors have been modified, and most of the metaphors suffer from possible cultural specificity (a problem discussed in another context by Keesing [36]). Overall, however, it is believed that these issues do not significantly detract from the utility of these metaphors, and that user testing will confirm this assumption.

The following interface design principles are suggested on the basis of the foregoing discussion:

Principle 6: Adapt metaphors that most closely fit the target domain.

Principle 7: Choose metaphors that minimise user learning of task interfaces.

Principle 8: Analyse all metaphoric entailments to ensure a high degree of fit with user perceptions and experience.

Principle 9: Identify and evaluate implicit metaphors which may be present in existing interfaces, as a result of the pervasiveness of metaphor in human cognition.

# 4.0 Conclusions

Among videogame developers, usability and universal access challenges are confronted afresh with every new title that enters development. However, finding the right balance between 'interface' and 'content' is not an easy task, because the challenge exists at several levels. Pagulayan *et al.* [58, p.890] go so far as to suggest that: "Thinking of interactivity as a wholly unique phenomenon in games can interfere with effective game design if it causes the designer to think about actions and sequences rather than empathizing with the way that their user will be thinking and feeling as they progress through the game. … the most important part of interactivity is not clicking a mouse or moving a joystick." This suggests that functional benefits such as efficiency and usefulness are not the only advantages to be obtained by adopting videogame interfaces. Improvements in user experience (or UX, [30]) are also likely to accrue when interface ideas are transferred from an application domain which is focused on delivering consumer satisfaction at the point of interaction. For their part, Harris and Harris [29] suggest that the challenge for interface designers resides not only in the individual user, but also in society at large, especially when safety issues are paramount. The evidence marshalled in this paper indicates that while the design-by-adaptation process is a potentially fruitful strategy for designing interfaces aimed at ensuring universal access, videogames do not represent a silver bullet, even for a high-affinity application domain such as VGE software. For every videogame interface design principle or mechanism that offers gains in universal access (e.g., the learning-by-doing style tends to suit those who find difficulty in absorbing and following formal procedural instructions) there are other design principles or mechanisms which represent losses in terms of universal access (e.g., the fast-reaction interfaces of many first-person shooter games tend to privilege younger users). As Harris and Harris [29, p.557] emphasise, the key to successful technology transfer is to ensure the "transfer of *appropriate* solutions to problems" (original emphasis). The detailed analysis of videogame interfaces presented in this paper represents one way of ensuring that appropriate interface ideas are distinguished from inappropriate ones.

A decade ago, van Dam [80] suggested that the steady year-on-year improvement in computer hardware performance was unmatched by similarly regular progress in user interface design, and that interface progress tended to follow a punctuated equilibrium model. VGE interfaces are currently entering a period of significant reappraisal, as this type of software evolves from its traditional role in handling 2D and 2.5D models of the world to presenting full 3D models. In this context, videogames represent a fertile nursery of ideas which can be raided to create improved 3D interfaces, not only for VGEs, but also for other applications in which users are required to explore 3D virtual environments. In 3D virtual space, it is no longer enough for software to provide sophisticated analytical functionally; software also needs to be accessible and usable. The core argument of this paper is that the goal of universal access may be advanced by adapting suitable interface ideas and mechanisms from a category of software in which broad appeal and usability have become benchmarks of success.

## Acknowledgements

We wish to thank conference participants who provided comments on an original version of this paper, which was presented at the International Conference on Virtual Geographic Environments, held at the Chinese University of Hong Kong, January 2008, and also to three anonymous referees for their incisive and helpful comments on the journal version of the paper.

## References

[1] Andrienko, N and Andrienko, G. (2003) Informed spatial decisions through coordinated views, *Information Visualization*, **2**, 270-285.

[2] Angus, I. G. & Sowizral, H. A. (1995) Embedding the 2D interaction metaphor in a real 3D virtual environment, Proceedings SPIE Vol. 2409, pp.282-293, S. S. Fisher, J. O. Merritt & M. T. Bolas (eds) *Stereoscopic Displays and Virtual Reality Systems II*.

[3] Bandelloni, R., Berti, S. & Paternò, F. (2005) Analysing trans-modal interface migrations, Proceedings INTERACT '05, Roma, *Lecture Notes in Computer Science*, 3585, pp.1071-1074 (Berlin: Springer-Verlag).

[4] Banks, R. & Wickens, C. D. (1997) Commanders' display of terrain information: manipulations of display dimensionality and frame of reference to support battlefield visualization, Technical Report ARL-97-12/FED-LAB-97-2, US Army Research Laboratory, Aberdeen, MD.

[5] Barr, P., Biddle, R. & Noble, J. (2002) A taxonomy of user-interface metaphors, in: S. Jones & M. Masoodian (eds) Proceedings of the SIGCHI-NZ Symposium on Computer-Human Interaction.

[6] Barr, P., Khaled, R., Noble, J. & Biddle, R. (2005) A taxonomic analysis of user-interface metaphors in the Microsoft Office Gallery project, Proceedings of the 6th Australian User Interface Conference (AUIC2005), pp.109-117.

[7] Bolt, N. & Tulathimutte, T. (2009) Researching videogames the UX way, www.boxesandarrows.com/view/research-video, 22 January 2009 (last accessed 12 August 2009).

[8] Bowman, D. A. (1999) *Interaction Techniques for Common Tasks in Immersive Virtual Environments*, Doctoral Thesis, Georgia Institute of Technology.

[9] Bowman, D. A., Kruijff, E., LaViola, J. J. & Poupyrey, I. (2004) *3D User Interfaces: theory and practice* (Addison-Wesley Professional).

[10] Burigat, S. & Chittaro, L. (2007) Navigation in 3D virtual environments: effects of user experience and location-pointing navigation aids, *International Journal of Human-Computer Studies*, 65, pp.945-958.

[11] Butz, A. & Krűger, A. (2003) A generalized peephole metaphor for augmented reality and instrumental environments, Proceedings of the International Workshop on Software Technology for Augmented Reality Systems (STARS).

[12] Calomeni, A. & Celes, W. (2006) Assisted and automatic navigation in black oil reservoir models based on probabilistic roadmaps, Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, pp.175-182.

[13] Card, S. K., Mackinlay, J. D. & Robertson, G. G. (1990) The design space of input devices, Proceedings CHI '90, pp.117-124 (New York: ACM Press.

[14] Carroll, J. M., Mack, R. L. & Kellogg, W. A. (1988) Interface metaphors and user interface design, in: M. Helander (ed.) *Handbook of Human-Computer Interaction*, pp.67-85 (North-Holland: Elsevier Science Publishers).

[15] Chittaro, L., Ranon, R. & Ieronutti, L. (2003) Guiding visitors of Web3D worlds through automatically generated tours, in: Proceedings of Web3D 2003, pp.27-38 (New York: ACM Press).

[16] Clanton, C. (2000) Lessons from game design, in: E. Bergman (ed.) *Information Appliances and beyond: interaction design for consumer products* (San Fransisco, CA: Morgan Kaufman Publishers).

[17] Clarke-Wilson, S. (1998) Applying game design to virtual environments, in: C. Dodsworth (ed.) *Digital Illusions: entertaining the future with high technology*, pp.229-239.Reading, MA: Addison Wesley).

[18] Darken, R. P. & Sibert, J. L. (1996) Wayfinding strategies and behaviours in large virtual worlds, Proceedings VHI '96, pp.142-149. (New York: ACM Press).

[19] Dourish, P. (2001) *Where The Action Is: The foundations of embodied interaction* (Cambridge MA: MIT Press).

[20] Erickson, T. D. (1990) Working with interface metaphors, in: B. Laurel (ed.) *The Art of Human-Computer Interface Design*, pp.65-73 (Reading, MA: Addison Wesley).

[21] Franck, A. U. (1993) The use of geographical information systems: the user interface is the system, in: D. Medyckyj-Scott & H. Hearnshaw (eds) *Human Factors in Geographical Information Systems*, pp.3-14 (London: Belhaven Press).

[22] Fröhlich, B., Barrass, S., Zehner, B, Plate, J. & Göbel, M. (1999) Exploring geo-scientific data in virtual environments, *IEEE Visualization, Proceedings of the Conference on Visualization '99: celebrating 10 years*, pp.169-173 (Los Alamos, CA: IEEE Computer Society Press).

[23] Galyean, T. A. (1995) Guided navigation of virtual environments, *Symposium on Interactive 3D Graphics, Proceedings of the 1995 Symposium on Interactive 3D Graphics*,103-4, 210.

[24] Guo, D., Chen, J. and MacEachren, A. M. (2006) A visualization system for space-time and multivariate patterns (VIS-STAMP), *IEEE Transactions on Visualization and Computer Graphics*, **12**(6), 1461-1474.

[25] Guy, B., Bidwell, N. J. & Musumeci, P. (2005) GamePlan: serious gaming for place making, Proceedings of the Second Australian Conference on Interactive Entertainment, pp.252-255.

[26] Hacklay, M. E. (2002) Virtual reality and GIS, in: P. Fisher & D. J. Unwin (eds) *Virtual Reality in Geography*, pp.47-57 (London: Taylor & Francis).

[27] Haik, E., Barker, T., Sapsford, J. and Trainis, S. (2002) Investigation into effective navigation in desktop virtual interfaces, *3D Technologies for the World Wide Web, Proceedings of the seventh International Conference on 3D Web Technology*, ACM Press, New York, pp.59-66.

[28] Hanson, A. and Wernert, E. (1997) Constrained 3D navigation with 2D controllers, *IEEE Visualization*, 175-182.

[29] Harris, D. & Harris, F. J. (2004) Evaluating the transfer of technology between application domains: a critical evaluation of the human component in the system, *Technology in Society*, 26, pp.551-565.

[30] Hassenzahl, M. & Tractinsky, N. (2006) User experience -- a research agenda, *Behaviour and Information Technology*, 25(2), pp.91-97.

[31] Hughes, S. B. (2006) Coordination strategies for assisted viewpoint interaction, Proceedings of the 2006 IEEE Symposium on 3D User Interfaces (3DUI'06), pp.17-24.

[32] Jacko, J. A. & Hanson, V. L. (2002) Universal access and inclusion in design, *Universal Access in the Information Society*, 2, pp.1-2.

[33] Jacko, J. A. & Vitense, H. S. (2001) A review and reappraisal of information technologies within a conceptual framework for individuals with disabilities, *Universal Access in the Information Society*, 1, pp.56-76.

[34] Jennett, C., Cox, A. L., Cairns, P., Dhoparee, S., Epps, A., Tijs, T. & Walton, A. (2008) Measuring and defining the experience of immersion in games, *International Journal of Human-Computer Studies*, 66, pp.641-661.

[35] Joiner, D. (1998) Real interactivity in interactive entertainment, in: C. Dodsworth (ed.) Digital Illusion: Entertaining the Future with high technology, pp.151-159 (New York: ACM Press).

[36] Keesing, R. M. (1985) Conventional metaphors and anthropological metaphysics: the problematic of cultural translation, *Journal of Anthropological Research*, 41(2), pp.201-217.

[37] Kitamura, Y, Ogata, S. & Kishino, F. (2002) A manipulation environment of virtual and real objects using a magnetic metaphor, Proceedings of the ACM Symposium on Virtual Reality Software and Technology.

[38] Kruger, W., Bohn, C-A., Froblich, B., Schuth, H., Strauss, W. & Wesche, G. (1995) The responsive workbench: a virtual work environment, *Computer*, 28(7), pp.42-48.

[39] Kuhn, W. (1993a) Various perspectives on interface metaphors, Proceedings of the Fourth European Conference and Exhibition on GIS, EGIS '93, pp.456-463. Genoa, Italy.

[40] Kuhn, W. (1993b) Metaphors create theories for users, in: A. U. Franck & I. Campari (eds) *Lecture Notes in Computer Science*, 716, pp.366-376.

[41] Leclerc, Y. G., Reddy, M., Iverson, L. & Bletter, N. (1999) Digital Earth: building the new world, www.ai.sri.com/~reddy/pubs/vsmm99/ (accessed 19 November 2007).

[42] Lewis, M. & Jacobson, J. (2002) Game engines in scientific research, *Communications of the ACM*, 45(1), pp.31.

[43] Li, T.-Y., & Ting, H.-K. (2000) An intelligent user interface with motion planning for 3D navigation, Proceedings of the IEEE VR 2000 Conference, pp.177-184.

[44] Manninen, T. (2003) Interaction manifestations in multi-player games, in: G. Riva, F. DavideW. A. Ijsselsteijn (eds) *Being There: concepts, effects and measurement in synthetic environments*, pp.295-304 (Amsterdam: Ios Press).

[45] Manninen, T. (2004) *Rich Interaction Model for Game and Virtual Environment Design*, Dissertation, Department of Information Processing Science, University of Oulu, Finland. herkules.oulu.fi/isbn9514272544/isbn9514272544.pdf (Accessed 10 November 2007)

[46] Mark, D. M. (1992) Spatial metaphors for human-computer interaction, Proceedings of the Fifth International Conference on Spatial Data Handling, Charleston, Vol. 1, pp.104-112.

[47] Marsh, S. L., Dykes, J. & Attilakou, F. (2006) Evaluating a visualization prototype with two approaches: remote instructional vs. face-to-face exploratory, *Proceedings of Information Visualization (IV '06)*.

[48] Masliah, M. R. & Milgram, P. (2000) Measuring the allocation of control in a 6-degree-of-freedom docking experiment, CHI 2000 Papers, pp.25-32.

[49] McCormick, E. P., Wickens, C. D., Banks, R. & Yeh, M. (1998) Frame of reference effects on scientific visualization subtasks, *Human Factors*, 40(3), pp.443-451.

[50] Meyer, M. (2000) *ALCOVE: Design and implementation of an object-centric virtual environment*, Master of Science Thesis, Passadena, CA: California Institute of Technology.

[51] MicroImages (2006) Floating North Arrow in 3D simulation, www.microimages.com/documentation/cplates/62NorthArrow.pdf (Accessed 21 August 2009).

[52] Miesenberger, K., Searle, G., Ossmann, R., Archambault, D. & Holzinger, A. (2008) More Than Just a Game: Accessibility in Computer Games, in: *Human-Computer Interaction and Usability for Education and Work*. pp.247-260,
https://online.tu-graz.ac.at/tug_online/voe_main2.getVollText?pDocumentNr=95371&pCurrPk=40168.

[53] Milgram, P. & Colquhoun, H. W. (1999a) A framework for relating head-mounted displays to mixed reality displays, *Proceedings of the Human Ergonomics Society 43rd Annual Meeting*, pp.1177-1181.

[54] Milgram, P. and H. Colquhoun (1999b) A Taxonomy of Real and Virtual World Display Integration, in: Y. O. a.H. Tamura (ed.) *Mixed Reality - Merging Real and Virtual Worlds*, pp.1-16 (Berlin: Springer Verlag).

[55] Miliano, V. (1999) Unrealty: application of a 3D game engine to enhance the design, visualization and presentation of commercial real estate, VSMM99.

[56] Monmonier, M. (1994) Graphic narratives for analysing environmental risk, M. Monmonier & A. M. McEachren (eds) *Visualization in Modern Cartography*, (London: Pergamon Press).

[57] Olive, J., Thouvenin, I. & Lenne, D. (2005) A new knowledge visualization and manipulation metaphor for collaborative and immersive environment, Proceedings of ENACTIVEOS, 2nd International Conference on Enactive Interfaces, Genoa, Italy.

[58] Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L. & Fuller, T. (2003) User-centred design in games, in: Jacko, J. A. & Sears, A. (eds) *The Human-Computer Interaction Handbook: Fundamentals, evolving technologies and emerging applications*, pp.883-906 (London: Lawrence Erlbaum Asoociates, Publishers).

[59] Paternò, F., Santoro, C. & Scorcia, A. (2008) Evaluating migratory user interfaces, in: E. L-C. Law, N. Bevan, G. Christou, M. Springett & M. Lárusdóttir (eds) *Proceedings of the International Workshop on Meaningful Measures: Valid Useful User Experience Measurement (VUUM)*, pp.79-85, June 18, Reykyavik, Iceland.

[60] Plumlee, M. and Ware, C. (2003) An evaluation of methods for linking 3D views, *Proceedings ACM SIGGRAPH 2003, Symposium on Interactive 3D Graphics*, pp.193-201, ACM Press, New York.

[61] .61 Powell, A. (2005) Composite metaphor, games and interface, Proceedings of the Second Australasian Conference on Interactive Entertainment, Sydney, Australia,, pp. 159-162.

[62] Raybourn, E. M. & Bos, N. (2005) Design and evaluation challenges of serious games, Proceedings CHI '05, pp.2049-2050 (New York: ACM Press).

[63] Rhyne, T-M. (2002) Computer games and scientific visualization, *Communications of the ACM*, 45(7), pp.40-44.

[64] Savidis, A. & Stephanidis, C. (2004) Unified user interface development: the software engineering of universally accessible interactions, *Universal Access in the Information Society*, 3, pp.165-193.

[65] Sayers, H. M. (2004) Desktop virtual environments: a study of navigation and age, *Interacting with Computers*, 16, pp.939-956.

[66] Sayers, H. M., Wilson, S. & McNeil, M. D. J. (2004) Navigational tools for desktop virtual environment interfaces, *Virtual Reality*, 7(3-4), pp.131-139.

[67] Schmitz, M., Baus, J. & Schmidt, S. (2006) Towards anthropomorphized objects: a novel interaction metaphor for instrumented spaces, w5.cs.uni-sb.de/publication/file/184/AnthropomorphizedObjects_CRV.pdf (accessed 10 November 2007).

[68] Schneiderman, B. & Maes, P. (1997) Direct manipulation vs. interface agents, *Interactions,* 4(6), pp.42-61.

[69] 69. Shepherd, I. D. H. (2000)*Key Skills: teaching and learning for transfer* (London: DfEE, and Cheltenham: Geography Discipline Network).

[70] Shepherd, I. D. H. (2002) It's only a game: using interactive graphics middleware to visualise historical data, *Society of Cartographers Bulletin*, 36(2), pp.51-55.

[71] Shepherd, I. D. H. (2008) Travails in the third dimension: a critical evaluation of three-dimensional geographical visualization', in: M. Dodge, M. McDerby & M. Turner (eds) *Geographic Visualization: Concepts, Tools and Applications* , Chapter 10, pp.199-222 (John Wiley).

[72] Shepherd, I. D. H. (2009) From Geography Department to Business School: strategies for transplanting GIS courses between disciplines, *Journal of Geography in Higher Education*, 33(Suppl.), pp.1-18.

[73] Shepherd, I. D. H. & Bleasdale-Shepherd, I. D. (2009a) Videogames: the new GIS?, in: H. Lin & M. Batty (eds) *Virtual Geographic Environments*, chapter 22. pp.311-344 (Science Press).

[74] Shepherd, I. D. H. & Bleasdale-Shepherd, I. D. (2009b) 'Experiential Engagement with VGEs: What can we learn from videogames? Proceedings of the ISDE 6th International Symposium on Digital Earth, Beijing.

[75] Steinicke, F., Ropinski, T. & Hinrichs, K.. (2005) Multimodal interaction metaphors for manipulation of distant objects in immersive virtual environments, WSCG2005, the 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision.

[76] Stoakley, R., Conway, M. & Pausch, R. (1995) Virtual reality on a WIM: interactive worlds in miniature, in: Proceedings of CHI 95, pp.265-272 (New York: ACM Press).

[77] Thomas, L. C., Wickens, C. D. & Merlo, J. (1999) Immersion and battlefield visualization: frame of reference effects on navigation tasks and cognitive tunnelling, Technical Report ARL-99-3/FED-LAB-99-2, US Army Research Laboratory, Aberdeen, MD.

[78] Tory, M, Kirkpatrick, A. E., Atkins, S. and Möller, T. (2006) Visualization task performance with 2D, 3D, and combination displays, *IEEE Transactions on Visualization and Computer Graphics*, **12**(1), 2-13.

[79] Treinish, L. (1998) Task-specific visualization design: a case study in operational weather forecasting, *IEEE Visualization, Proceedings of the Conference on Visualization '98*, 405-408.

[80] van Dam, A. (1997) Post-WIMP user interfaces, *Communications of the ACM*, 40(2), pp.63-67.

[81] Ware, C. and Osborne, S. (1990) Exploration and virtual camera control in virtual three dimensional environments, *Computer Graphics*, 24(2), pp.175-183.

[82] Wickens, C. D. (1999) Frame of reference for navigation, in: D. Gopher &. A. Koriat (eds) *Attention and Performance*, (Vol. 16), pp.113-143 (Orlando, FL: Academic Press).

[83] Woods, S. (2004) Loading the dice: the challenge of serious videogames, *Game Studies: The International Journal of Videogame Research*, 4(1), www.gamestudies.org/0401/woods/ (accessed 19 November 2007).

[84] Dickey, M. D. (2006) Game design narrative for learning: appropriating adventure game design narrative devices and techniques for the design of interactive learning environments, *Educational Technology Research and Development*, 54(3), pp.245-263.

**Table 1.** Differences between videogames and productivity software (adapted from Pagulayan *et al*., 2003).

| **Characteristic** | **Videogames** | **Productivity Applications** |
|---|---|---|
| Purpose | Pleasure, fun, entertainment | Tools/tasks/productivity |
| Goals | Designed in | Defined by users |
| Software choice | Many alternatives | Few alternatives |
| Consistency | Low (variety of experience important) | High (familiarity important) |
| Constraints | Deliberately imposed | Avoided/designed out |
| Non-visual cues | Create mood | Denote functions (represent data) |
| Point of view | Frequently built in | Rare |
| Influential purchasers | Consumers | Organisations |
| Innovation | Welcomed | Cautionary approach |
| Input devices | Varied/platform-specific | Standard/cross-platform |

**Table 2.** Degrees of freedom (DOF) used in 3D videogames. (Note: All mice and joysticks are analog devices)

| DOF | Game Title | Interface description |
|---|---|---|
| 6 | *Descent*<br>(PC)<br>[First-person shooter (FPS)] | 2 keys mapped to 'pitch' rotation.<br>2 keys mapped to ‘yaw’ rotation.<br>2 keys mapped to ‘roll’ rotation.<br>2 keys mapped to 'forward/backward' translation.<br>2 keys mapped to 'left/right' translation.<br>2 keys mapped to 'up/down' translation.<br>(Note: All above key mappings are relative to current viewpoint rotation.)<br>This interface is cumbersome and, in combination with a zero-gravity environment, confusing. The roll, up/down and left/right controls are typically not used by players, due to a shortage of fingers. |
| 5 | *Super Mario 64*<br>(Console, underwater)<br>[Third-person platformer] | 2 DOF of joystick mapped to vertical/lateral rotation, relative to current orientation of viewpoint.<br>1 button mapped to 'forward' translation, relative to the current viewpoint orientation.<br>This interface has essentially just 1 DOF (roll). It drastically simplifies the controls the user must master. |
| 4 | *Half-Life 2*<br>(PC, on land)<br>[First-person shooter (FPS)] | 2 DOF of mouse map directly to 2 rotational DOF of viewpoint.<br>2 keys mapped to 'forward/backward' translation in horizontal plane, relative to current viewpoint yaw.<br>2 keys mapped to 'left/right' translation in horizontal plane, relative to current viewpoint yaw.<br>1 key mapped to 'jump', providing momentary vertical motion.<br>This is the de-facto standard control interface for modern FPSs.<br>Control interface eschews the 2 least significant DOF (i.e., roll and up/down) for the sake of simplicity, focusing on the ergonomics of controller usage. |
| 3 | *Pikmin*<br>(Console)<br>[Real-time strategy (RTS) game] | 2 DOF of joystick mapped to translation of player's avatar within horizontal plane, which the viewpoint follows.<br>2 buttons mapped to increase/decrease elevation of viewpoint with respect to horizontal plane. |
| 2 | *Echochrome*<br>(Console)<br>[Puzzle] | 2 DOF of mouse map directly to 2 rotational DOF of viewpoint, which orbits the geometry comprising each of the game's levels. |
| 1 | *The Crimson Room*<br>(PC)<br>[Puzzle] | Clicking at screen borders causes translation + rotation of the viewpoint along a pre-defined path.<br>Mouse cursor free to move within current view and click on objects therein. |

**Table 3.** Degrees of viewpoint-focus coupling in videogames. (Note: All mice and joysticks are analog devices)

| **Degree of coupling** | **Game** | **Interface Description** |
|---|---|---|
| Complete | *Half-Life 2*<br>(PC)<br>[First-person shooter] | 2 DOF of the mouse map directly to 2 rotational DOF of viewpoint.<br>2 keys mapped to 'forward/backward' translation in horizontal plane, relative to current viewpoint yaw.<br>2 keys mapped to 'left/right' translation in the horizontal plane, relative to current viewpoint yaw.<br>1 key mapped to 'jump', providing momentary vertical motion when on land and sustained upwards motion when under water. |
| Strong | *Jak & Daxter*<br>(Console)<br>[Third-person platformer] | 2 DOF of one joystick mapped to motion of in-game avatar in horizontal plane, relative to current viewpoint yaw.<br>Viewpoint automatically follows at a semi-rigidly fixed distance behind avatar, turning as needed to look toward the avatar.<br>2 DOF of one joystick optionally mapped to (inertial + damped) velocity of 2 rotational DOF of the viewpoint, orbiting around avatar.<br>Elevation of viewpoint coupled with distance from avatar, so view is low-and-close or high-and-far. |
| Medium | *God of War*<br>(Console)<br>[Third-person platformer] | 2 DOF of one joystick mapped to motion of in-game avatar in horizontal plane, relative to current viewpoint yaw.<br>Viewpoint moves along pre-defined path (i.e., it has 1 DOF), its angle and distance from avatar tuned to best suit each gameplay scenario. |
| Weak | *Ico*<br>(Console)<br>[Third-person action] | 2 DOF of one joystick mapped to motion of in-game avatar in horizontal plane, relative to current viewpoint yaw.<br>Various buttons mapped to context-dependent actions (such as grabbing, climbing up onto or leaping away from a narrow ledge on a wall).<br>The viewpoint remains mostly fixed in place for the duration of each 'scene', rotating only slightly to follow the character's motion.<br>It will also move slightly along a 1 DOF pre-defined path as the avatar moves, but will remain at a significant distance from it.<br>The viewpoint makes a dramatic transition to a new position and orientation as the avatar exits one scene and enters another.<br>These moments of transition are pre-defined and chosen so as to avoid interfering with the player's goals. |

| | | |
|---|---|---|
| Marginal | *Rez*<br>(Console)<br>[Rhythm action] | 2 DOF of one joystick mapped to a cursor, which is very slightly coupled to viewpoint rotation.<br>Aside from this, the position and orientation of player's avatar and viewpoint are determined by a pre-defined path along which they travel at a fixed rate. |
| None | *Resident Evil*<br>(Console)<br>[Survival horror] | 1 DOF of a joystick mapped to yaw of player's avatar.<br>1 DOF of a joystick mapped to forward/backward motion of player's avatar, relative to its current yaw in horizontal plane.<br>This is widely agreed to be a very poor interface for avatar navigation.<br>The viewpoint remains motionless for the duration of each 'scene'.<br>The viewpoint transitions to a new position and orientation, during a 'fade to black' moment, as the avatar exits one scene and enters another. |

**Table 4.** A selection of interface metaphors in 3D videogames. (Note: All mice and joysticks are analog devices)

| **Control Metaphor** | **Game** | **Metaphor Description (General, then game-specific)** |
|---|---|---|
| **Running** | This metaphor relates to the everyday experience of moving around at various speeds as a bipedal human. Typically, rotation of the head and motion of the feet are modelled in the horizontal plane. | |
| | *Half-Life 2* (PC) [First-person shooter] | 2 DOF of mouse map directly to 2 rotational DOF of viewpoint.<br>2 keys mapped to 'forward/backward' translation in horizontal plane, relative to current yaw of viewpoint.<br>2 keys mapped to 'left/right' translation in horizontal plane, relative to current yaw of viewpoint.<br>1 key mapped to 'jump', providing momentary upwards vertical motion. |
| | *Jak & Daxter* (Console) [Third-person platformer] | 2 DOF of one joystick mapped to motion of in-game avatar in horizontal plane, relative to current yaw of viewpoint.<br>Viewpoint automatically follows at a semi-rigidly fixed distance behind avatar, turning as needed to look in direction of avatar.<br>2 DOF of one joystick optionally mapped to (inertial and damped) velocity of 2 rotation DOF of viewpoint, orbiting around avatar.<br>Viewpoint elevation coupled with distance from avatar, so view is low-and-close or high-and-far. |
| | *Resident Evil* (Console) [Survival horror] | 1 DOF of a joystick is mapped to yaw of player's avatar.<br>1 DOF of a joystick is mapped to forward/backward motion of player's avatar, relative to its current yaw, in horizontal plane.<br>The viewpoint remains motionless for the duration of each 'scene'.<br>The viewpoint transitions to a new position and orientation, during a 'fade to black' moment, as the avatar exits one scene and enters another. |
| **Driving** | This relates to the everyday experience of driving a car, and tends to match the physical constraints of automobile motion. | |
| | *GTA3* (Console) [Third-person action] | 1 DOF of single joystick maps to steering left/right<br>1 digital button maps to forwards acceleration<br>1 digital button maps to forwards deceleration<br>2 digital buttons each rotate viewpoint 90 degrees to left/right, when held. When both are held, viewpoint is rotated |

| | | |
|---|---|---|
| | | 180 degree. This allows the player to quickly glance to sides and rear of the car. |
| | *Halo*<br>(Console)<br>[Third-person shooter] | 2 DOF of one joystick mapped to (inertial and damped) velocity of 2 rotation DOF of viewpoint, orbiting around avatar.<br>1 DOF of single joystick maps to forwards/backwards motion of avatar relative to current yaw of viewpoint.<br>AI automatically steers and accelerates/reverses car in order to reorient it to match the required direction. |
| **Pushing & pulling** | Again, this relates to the everyday experience of pushing and pulling objects. | |
| | *Zelda*<br>(Console)<br>[Role Playing Game] | In certain puzzles, the player is required to push and/or pull heavy stone blocks into specific arrangements.<br>This is accomplished by attaching to a block by holding a button, and moving avatar in the appropriate viewpoint-relative direction.<br>This direction is provided by an analogue joystick and is automatically quantized to be a multiple of 90 degrees. |
| | *Ico*<br>(Console)<br>[Third-person adventure] | At one point in the game, the player is required to move a platform up a tall lift shaft.<br>The lift is operated by means of Archimedes' screw principle.<br>The player attaches their avatar to a level by holding a button and, while attached, rotates one analogue joystick in circles (clockwise to lower the lift, anti-clockwise to raise it). |
| | *Elebits*<br>(Console)<br>[Action] | In this game, the player is charged with finding hordes of tiny hiding creatures.<br>To reveal the creatures, the player grabs and moves objects under or inside which these creatures are ensconced.<br>To move an object, the player attaches an energy beam to it by pointing at it with the *Wii* controller pointing device and holding a button. They then pull the object by pointing the controller away from the object in the desired direction. |
| **Flying** | Flight (of a character or flying vehicle) is a relatively common metaphor in videogames. | |
| | *Ace Combat 6*<br>(Console) [Flight Action] | One analogue joystick used to control banking and pitch of a jet fighter.<br>Left and right 'triggers' affect throttle.<br>Second analogue joystick rotates third-person viewpoint about the plane, with a smooth but exaggerated motion designed to allow the player to quickly look 90 degrees to the sides. |

| | | |
|---|---|---|
| | *Super Mario 64* (Console) [Third-person platformer] | When flying, Mario is controlled with a single joystick, which controls banking and pitch.<br>Unlike a powered plane, avatar acts more like a glider, where pitch trades height for speed. |
| | *Star Fox 64* (Console) [Flight Action] | Flight in this game is simplified to suit the shoot-em-up style of gameplay.<br>The player uses single joystick to move their ship up/down/left/right within the viewport, whilst always traveling forwards along a fixed 'corridor' through the level, and at a fixed rate. |
| **God** | This metaphor is familiar to videogame players. It treats the player as a god-like entity, controlling a large space (sometimes a whole world) from above. | |
| | *Sim City 4* (PC) [Simulation] | This game presents an orthographically projected city to the player.<br>The player navigates by zooming in/out with the mouse wheel and by moving mouse to the edges of the screen (if they move mouse to top of the screen, the viewpoint moves North, and so on). |
| | *Black & White 2* (PC) [Real-time strategy] | This game presents an island to the player.<br>To navigate, the player may translate the viewpoint by clicking-and-dragging on the landscape with a disembodied hand (controlled via the mouse) floating a few metres above the virtual ground.<br>The player zooms in/out with the mouse wheel, and rotates the viewpoint in fixed increments with cursor keys. |
| | *Homeworld* (PC) [Real-time strategy] | This game is set in space. Although rendered in full 3D, the action takes place within a 2D ecliptic plane.<br>The player's view into the world is constrained accordingly – they may rotate the view in any direction and zoom in or out, but may only translate the viewpoint's position within the ecliptic plane. |

**Table 5a.** Metaphor-based interaction mechanisms for VGEs.

| **Metaphor** | **Behaviour** |
|---|---|
| Free flight | Move around 3D environments using completely free movement |
| Tethered exploration | Explore the neighbourhood of a selected location or object, changing bearing and elevation |
| Terrain following | Explore surface features across a landscape (cf. Google Earth) or polyhedral object |
| Border patrol | Explore linear or polygonal objects by moving along them or around their boundary at a fixed distance |
| Tracking | Follow a moving object from a fixed distance and a given direction |
| Guided missile/Arrow | Move along a path from viewpoint to target |
| Grand tour | Move along a path connecting a succession of viewpoints |

**Table 5b.** Some entailment issues for proposed interaction metaphors.

| **Metaphor** | **Entailment issues** |
|---|---|
| Free flight | • Less familiar in cultures without film/TV |
| Tethered exploration | • Movement also permitted above the ground surface<br>• Viewpoint attached to end of tether (egocentric) rather than at some other location (exocentric) |
| Terrain following | • Less familiar in cultures without aircraft or film/TV |
| Border patrol | |
| Tracking | • Fixed-distance tracking perhaps less familiar among hunting cultures in densely vegetated regions |
| Guided missile / Arrow | • Path may not be a straight line (cf. arrow)<br>• Acceleration profile along path is different<br>• No impact or damage on arrival at target |
| Grand tour | • Possibly culture-specific (18th/19th century Europe) |