Comparing 3D Content Creation Interfaces in Two Virtual Worlds: *World of Warcraft* and *Second Life*

Victoria McArthur *York University (Canada)*
Robert J. Teather *York University (Canada)*
Wolfgang Stuerzlinger *York University (Canada)*

Abstract

In this article we compare the content creation systems of two popular virtual worlds: *World of Warcraft* and *Second Life*. We then discuss recommendations for 3D content creation systems based on current trends in 3D user interface research. We hypothesize that by designing 3D content creation systems that follow these recommendations, virtual world economies based on custom content creation (e.g., *Second Life*) may be transformed, as more people will be able to create and modify content.

Key Words: 3D user interfaces, virtual worlds, content creation, metaverse, v-commerce

1. Introduction

Virtual worlds such as *Second Life* (Linden Research 2009b) have become a place for virtual commerce, community, and learning. Through the use of virtual worlds, companies such as IBM and Sun Microsystems have experimented with new forms of communication, collaboration and economic activity (Brandon 2007).

Second Life differs from other three-dimensional (3D) virtual worlds in that its users can conduct monetary business with one another directly through the use of Linden Dollars (L$), *Second Life*'s own virtual currency. This differs from economic systems in other popular virtual worlds, such as Blizzard's *World of Warcraft* (Blizzard Entertainment 2009b), in that the direct conversion between real world currency and virtual currency is a feature only found in Second Life (Rymaszewski et al. 2007). This, in combination with the ability to create custom content within *Second Life*, is one reason why companies such as IBM and Sun have invested in developing content for, and having a presence within, these environments (Brandon 2007).

The economy of *World of Warcraft* is oversaturated with 'skilled' item producers. The items crafted by players (weapons, armour, potions, etc.) are all generated using familiar 2D interaction metaphors, such as selecting items from menus and lists. The resultant product is a virtual object that has been pre-modeled by the game developers with predetermined characteristics. These items are not customizable by the user. No real-world skills are required to be a producer in this environment - only time and patience.

Conversely, creating content in *Second Life* is much more difficult. Users must learn how to create 3D objects, apply textures, and script the object's behaviour. This requires a broad range of real-world skills including 3D modeling and programming. The producers in this system have a monopoly through their skills, which is the main reason they are in business. To some degree, this parallels skill-driven real world economies.

Recent research in 3D user interfaces has produced more user friendly content creation systems, e.g. (Oh and Stuerzlinger 2005; Teather and Stuerzlinger 2008b). Commercial products such as Google's *Sketchup* (2007), or games like Electronic Arts' *Spore* (2008) are representative of these advances in 3D content creation. Early reviews of *Spore's Creature Creator,* the interface for building creatures described the experience as being like 'manipulating a virtual lump of clay on the screen' (Ocampo 2008). From a usability standpoint, the current model used by *Second Life* is fairly primitive and in-line with 'old-style' 3D modeling techniques available in high-end software such as Autodesk's *3D Studio Max* or *Maya* (2010). Phrased differently, these systems provide a user interface that is a thin layer over the math, and require a great deal of training and practice to acquire a high level of skill. We argue that it is the high level of skill required to use these tools that is one of the main factors that add value to the items created in *Second Life*.

Recent work in 3D user interface design has resulted in 3D content creation interfaces that are based on reflection about the capabilities of humans and the constraints reality poses on humans and their experience (Oh and Stuerzlinger 2005; Smith et al. 2001). Consequently, humans seldom construct real objects in full 3D, i.e. in free air, but rely on a variety of constraints (such as gravity, contact between objects, etc.) to aid construction tasks. Techniques that support these ideas essentially reduce the dimensionality of the task from full 3D to 2D or 2.5D. Not only is this is conceptually easier for users, especially novices, but also more readily fits the functionality of desktop input devices (e.g., the mouse). We will provide an overview of several concepts that have been empirically demonstrated to improve 3D object manipulation and 3D content generation, for novice users in particular.

We argue that using such constraints and adding user interfaces that provide a better match between human capabilities and online virtual environments would dramatically reduce the skill requirement to construct in-world items. We speculate that removing this skill requirement and making 3D products easier to build in *Second Life* may transform the current e-commerce model.

2. Background

In this section we provide an overview of related work on virtual world economies as well as an overview of related work on 3D content creation techniques.

2.1 Virtual world economies

Virtual worlds are ripe grounds for economic discussion. Outside of the virtual world itself, there is often a tension between the creation of the virtual world, and the economic cost of its maintenance. In the case of many Massively Multiplayer Online Role Playing Games (MMORPGs), players may purchase the original game from the developer or a retail store and may or may not have to pay a monthly subscription fee to play. Developers may also release expansion packs that offer new game content to players.

World of Warcraft, for example, currently costs $19.99 USD if purchased directly from the Blizzard Online store (Blizzard Entertainment 2009a). Players must then pay a monthly subscription fee of $14.99 USD to play the game. In addition to the original game, developer Blizzard has released two expansion packs, *The Burning Crusade* in early 2007 and *Wrath of the Lich King* in late 2008. These expansion packs can be purchased from the Blizzard Online store for $29.99 and $39.99 USD respectively. These expansion packs have dramatically altered the game since its initial release by adding new 3D content for users, new playable races, new regions, and improved graphics. Recently, Blizzard announced a third expansion, *Cataclysm*, targeted for a 2010 release.

Two other popular MMORPGs are Areanet's *Guild Wars* (Areanet 2009) and Jagex Ltd.'s *Runescape* (Jagex 2009). *Guild Wars* is similar to *World of Warcraft* in that the original game and subsequent expansions must be purchased to be played. *Guild Wars* differs from

World of Warcraft in that players are not charged a monthly subscription fee to play the game. *Runescape* differs from both of these games in that it is played in a web browser, rather than installed on the player's computer, and is also entirely free to play.

Second Life is based on an entirely different philosophy altogether. Firstly, although its participation is massively-multiplayer in nature, the creators of *Second Life* and members of its community are adamant about not referring to it as a game (Kalning 2007). This is mostly because, unlike the aforementioned environments, *Second Life* has no game-like elements. *Second Life* offers no quests, experience points, or rewards. Instead *Second Life* is a metaverse; a virtual world that is co-constructed by its inhabitants (McArthur 2008). It is for this reason that the terms 'social' and 'game-based' have emerged in order to differentiate between the two types of virtual worlds (Ducheneaut et al. 2009). Members of the community may choose to play roles via their avatars, or they may choose to just be themselves (Ducheneaut et al. 2009). There is no subscription required to join *Second Life* or to participate in the virtual world. However, residents who wish to own land must pay a monthly lease fee based on the amount of land they wish to own (Linden Research 2009a).

While game sales and subscription fees describe one aspect of economies associated with virtual worlds, even more interesting are the emergent economies found within. Virtual world economics are based on a combination of established real world business models in addition to emergent business models (Noam 2007). The unique nature of a virtual environment allows for some interesting potential business models. Noam presents eight business models for virtual worlds:

1. Marketing of Real World Goods & Services
2. Selling to Users (V-Commerce)
3. B2B Services to Virtual World Business Operators
4. New-Style Services
5. Media Content Distribution
6. The Resort Economy
7. Community Creation
8. Owning the Virtual World

Examples of the above business models can be seen in many of the most popular virtual worlds. In the context of this article, we are most interested in the second business model: *Selling to Users or V-Commerce* and how it relates to the content creation interfaces of these virtual worlds. In this business model, consumers purchase virtual items, such as a virtual shirt or furniture for their avatar. Avatars in 3D virtual worlds are three dimensional virtual bodies that act as a representative in-world that is controlled and maintained by the user.

As with real-world business models, *V-Commerce* is affected by supply and demand. In this work, we focus on the *supply* component of this business model as it pertains to content creation – in this case, virtual objects – in two exemplary virtual environments: *Second Life* and *World of Warcraft*. In our analysis of some of the goods and services offered in both environments, we discuss how their content creation systems impact the structure and stability of their economies.

2.2 3D content creation

To this day, few games support any kind of full-featured 3D object manipulation. This is at least in part due to the lack of suitable input devices as well as the lack of intuitive interaction techniques for these devices. One issue is that manipulating 3D objects requires handling six degrees of freedom (6DOF), i.e., there are three axes of movement and three axes of rotation for every object. To contrast, an example of a 2DOF problem is manipulating desktop icons with a mouse, as there are only two axes of movement. A large body of virtual reality (VR) research focuses on the development of efficient object manipulation techniques using 3D input devices such as trackers and wands (Bowman et al. 2004). A major motivation for this is that these devices allow the user to simultaneously position and orient a virtual object, and thus may provide a more efficient manipulation interface compared to input devices that control fewer DOFs.

In practice, most users are extensively familiar with 2D input devices, in particular the mouse. Moreover, all commercially successful 3D modeling systems use a mouse-based user interface. However, using a mouse for 3D interaction introduces the problem of mapping 2D mouse cursor motions into 3D operations. While several solutions have been proposed (Bier 1987; Conner et al. 1992), all require users to mentally translate 2D mouse movements into low-level 3D operations such as movement along, or rotation about, one axis at a time. This is unsuitable for naïve users. Alternatively, there is evidence that 2D input devices can outperform 3D devices for the most frequently used 3D positioning tasks in 3D scene construction (Teather and Stuerzlinger 2008b). This is done with 'intelligent' software techniques that map mouse movement to intuitive 3D object movement that conforms to real-world constraints and experience.

2.2.1 3D object selection and manipulation

Selection refers to the act of specifying a 3D object as the target for subsequent operations. These subsequent operations are often 3D manipulation tasks. General manipulation is a full 6DOF task, consisting of both translation (movement) and rotation of an object. In this section we discuss these two actions in the context of related work.

Previous work has developed taxonomies of 3D selection/manipulation techniques (Bowman et al. 1999; Poupyrev et al. 1998). Poupyrev et al. (1998) compared selection and manipulation with ray-casting and a virtual hand metaphor. Ray-casting techniques only require pointing at the intended object to select (and manipulate) it. Virtual hands typically require intersection of a hand avatar, or the users real hand with the intended object. Poupyrev's work suggested that there was no clear winner – each technique tested had advantages and disadvantages, depending on factors such as distance to the target, object size and visual feedback. Bowman et al. (1999) presented a study that compared several techniques created from basic 3D interaction components, and evaluated them in a selection and manipulation test-bed. Contrary to Poupyrev's results, they found that selection based on ray-casting was significantly faster than selection techniques requiring full 3D hand/cursor movement. For manipulation, they found that the degrees of freedom of the manipulation task had a significant effect on task completion time. In fact, they note that it dominated the results, with 2DOF (two degree of freedom) techniques significantly outperforming 3DOF techniques, on average. This suggests that 2D input devices such as the mouse are not only suitable for 3D manipulation tasks (such as constructing 3D geometry), but may be *very* well-suited to the task – if appropriate software techniques are used to map input to action.

Boritz and Booth (Boritz and Booth 1998; Boritz and Booth 1997) conducted a series of studies on 6DOF input devices for 3D interaction tasks. They first studied the use of 6DOF input devices for selection tasks (Boritz and Booth 1997). In their study, they compared stereoscopic to monoscopic display with and without head tracking, as well as different target positions. The presence of stereo display and head tracking provides additional depth cues, allowing viewers to more easily perceive spatial relationships between objects. This is widely believed to improve object selection and manipulation. A second study also considered orientation of the target (Boritz and Booth 1998), requiring users to dock a 3D cursor with a target, matching both position and orientation – a full 6DOF task. Both studies showed that stereo viewing significantly improved task completion time, but head tracking did not have an effect. The authors reason that their tasks required only minimal head movement after the initial discovery of target locations. They note that although positional error was reduced in the stereo viewing mode, display mode showed no significant difference between stereoscopic and monoscopic for *rotational* error. With the exception of the true 6DOF docking task in Boritz et al.'s second study (1998), the studies mentioned above used only three of the six afforded by the 6DOF input devices for manipulation as in all but the docking study, the 6DOF input device was only used for positioning, not orientation..

Other work points out that 2D input devices work well for 3D interaction when ray casting is used for selection and manipulation (Oh and Stuerzlinger 2005; Poupyrev et al. 1998; Smith et al. 2001; Ware and Lowther 1997). When using a mouse, ray casting is commonly used for 3D object selection. The position of the mouse cursor is the origin of the ray, which extends into the 3D scene. The ray is checked for intersections with objects in the scene, and the closest intersected object is selected (e.g., when clicked). This effectively constrains 3D selection to a 2D 'point and click' task, as one need only interact with the visible projections of all objects, which can be described completely by a 2D image. Ware and Lowther (1997) conjecture that users rarely wish to interact with totally occluded objects, and as ray-casting allows the user to pick any (even only partially) visible object, this is sufficient. Ware and Lowther's study found that a ray-casting based 2D selection technique using a cursor rendered to a single eye in a stereo display was more accurate than a 3D selection cursor/virtual hand.

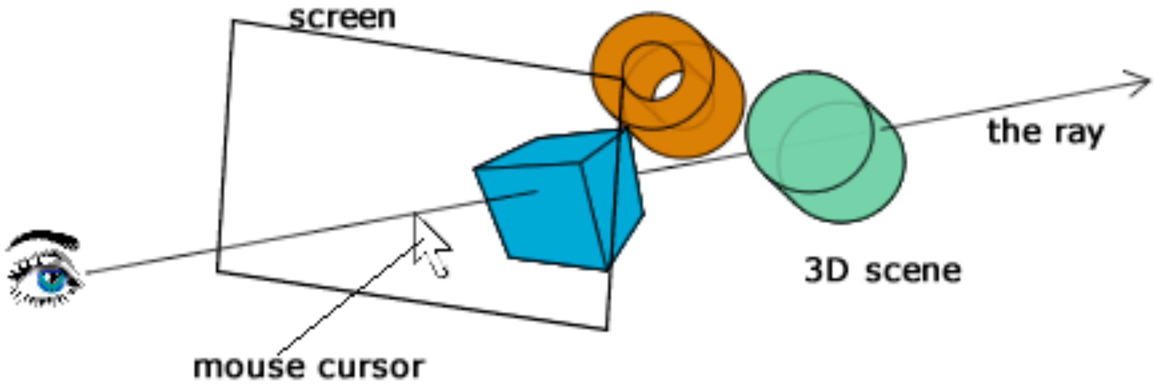

Figure 1: The ray is the infinite vector that originates at the camera/eye and passes through the mouse cursor and into the scene. It allows selection of all visible objects: those whose projections lie within the extents of the screen, and are not occluded by other objects in the scene. In 2D, this allows 'point and click' selection of

3D objects using a mouse by selecting the closest object to the viewer. In 3D systems, the actual eye and hand/cursor positions can be used, if they are known (i.e., if 3D trackers are used).

Manipulating 3D objects with a mouse is less straightforward than selection, since it is a 6DOF task, and the mouse only affords the simultaneous manipulation of two degrees of freedom. Thus, 2D input must be mapped to 3D operations. The mappings are frequently indirect and non-intuitive. We speculate that this is one reason for the lengthy learning curve common in existent 3D modeling software.

A common solution used in most modeling and commercial computer-aided design (CAD) systems is to use so-called '3D widgets' or handles (Conner et al. 1992), which separate the DOFs by explicitly breaking the manipulation down into its individual components. Small handles (arrows) are provided for movement along each of the three axes, and for each axis of rotation. Note that *Second Life* uses a variant of this approach for its 3D modeling system. These handles are usually complemented by three or four different simultaneous views of the same scene, typically an orthographic top view, two orthographic side views, and a perspective view. Bier's skitters and jacks technique (Bier 1987) provides a similar solution by interactively sliding a 3D cursor over objects in the scene via ray-casting, and attaching a transformation coordinate system to the object where it was positioned. Although commonly used, and easy to implement from a software perspective, the disadvantage of such manipulation techniques is that users need to mentally decompose every high-level movement task into individual operations along the axes of the object's local coordinate system – which do not necessarily align with the axes of the scene. Hence, high-level movement tasks such as 'put that object over there' are broken down into a series of more complex tasks such as 'move the object 10 units along the *x* axis, then 5 units along the negative *y* axis, and finally 10 units along the *z* axis'. These techniques are also prone to mode errors, i.e. movements along the wrong axis. These occur when a user accidentally performs the wrong operation because they thought the software was in a different mode. Mode errors are often the result of poor feedback mechanisms in the software.

A similar option, which results in an even lower-level interface is the use of direct coordinate entry (i.e., with a keyboard). While this can be useful for precision object positioning, it is even less direct than the use of widgets, and introduces the exact same problem of decomposing high-level conceptual movement tasks into low-level technical tasks.

Another approach is to constrain the movement of objects according to physical laws such as gravity and the inability of solid objects to inter-penetrate each other in reality. So-called 'semantic constraints' can also be used to limit object movement according to human expectations (Smith et al. 2001): e.g. chairs sit on the floor, and desk lamps sit on top of desks. Although this effectively deals with the issues presented by 3D widgets, this approach lacks generality, as it requires object-specific constraints to be designed a priori for each available type of object. This approach is thus unsuitable for general content creation provided by full-featured tools such as *3D Studio Max,* or the content creation interface present in *Second Life*. Note that such constraints may be suitable for games and game level editing software, as these typically support only a limited set of objects in a restricted environment.

A more general approach is based on the observation that in the real world virtually all objects are attached to other objects and hence remain in contact with other objects at all times (Oh and Stuerzlinger 2005). To achieve this, the movement algorithm uses the nearest surface occluded by the moving object to determine the current movement surface, while still avoiding collisions. An extension allows users to also move objects partially behind other objects. If an object is moved over the background, it moves in free space on a plane orthogonal to the viewer.

The result is that the object being moved always slides over the remainder of the scene in a very natural and predictable way that is consistent with results from recent visual perception research. With such an algorithm, the object being moved always remains in contact with other objects in the scene. Yet, it does not rely on the notion of gravity, i.e., one can move objects from the floor to walls or onto the ceiling and back. For efficiency, most of the computations are performed in graphics hardware.

A large number of games use a mouse for 3D *navigation* (e.g., *Doom3*, *Half-Life*, etc.), but very few games allow 3D *manipulation* of any degree. One of the few exceptions is Black & White 2 from Lionhead Studios (see http://www.lionhead.com/bw2), which allows movement of 3D objects in the game world using the mouse as a metaphorical hand. Clicking objects picks them up and holds them in-hand. The game's physics engine constrains objects to move according to user expectations when objects are released or thrown. However, orientation of objects is seldom, if ever, relevant to the game, and other than rotating the view around an object before grasping it, no facility is provided for rotating objects. Electronic Art's *Spore* is another example of a game that features a fairly advanced 3D content creation system – a key feature of the game. The player can create custom content in the form of alien life forms, buildings and vehicles. Adding parts to these constructs behaves similarly to the sliding algorithm described above (Oh and Stuerzlinger 2005). We speculate that this is likely partially responsible for the success of the content creation interface used.

Prior to discussing how these techniques impact the virtual economies of these two virtual worlds, we will first present an overview of some of the goods and services that are traded in-world, and how they are created.

2.3 3D content creation interfaces in virtual worlds

In both *World of Warcraft* and *Second Life*, a variety of goods and services can be traded among users. While many of these economical interactions are built into the game's interface, a number of emergent and un-supported exchanges occur as well. These are outlined in the detailed discussions of both of these virtual worlds. Note that we are only interested in the *player to player* economies of these environments. Thus, discussions about developer Blizzard's economic gain through monthly subscriptions and other goods and services are excluded here.

Presently, there are 14 trade skills in *WoW*. These are skills a player character can learn in order to produce goods and make money in-game from their sale. The goods the player creates can also be used to create weapons, armor, and magical items that can be used by the player themselves. These skills are divided into three categories: professions, gathering skills, and secondary skills (see Table 1). Players may learn at most two skills from either the professions or gathering skills but can learn as many of the secondary skills as they like. For example, a player may choose to learn blacksmithing and mining, or they may choose to learn two from the same category, such as herbalism and skinning. If a player wishes to learn a different skill from either of these two categories, they must first un-learn a profession or gathering skill, thus freeing up a slot. Players may, however, learn any or all of the secondary skills at any time.

Motivations for choosing these skills vary. For example, each of the gathering skills provides some of the materials necessary for yielding professional goods. Mining may supply a blacksmith with the copper necessary to forge copper chain pants or a copper claymore. Mining may also supply a jewel crafter with the silver necessary to craft an elegant silver ring. Since any two skills from these categories may be learned, players who choose two trade skills from the profession category must buy the materials they need directly from other players, or indirectly via the Auction House. Conversely, players may choose to learn two gathering skills and earn money by selling these raw materials via the Auction House. The Auction House is an in-game interface that allows characters to buy and sell any game items (e.g., dungeon loot, herbs, etc.). The price for these materials is set by the seller and auctions may last for up to 48 hours. Many players who focus on gathering skills exclusively, or players who take on new professions, may earn a lot of gold via the Auction House. Inexperienced players may inadvertently disrupt this process by setting the price on their auctions too low. Consequently, more experienced players may frequently watch the Auction House for low-priced materials, purchase them, and place them back in the Auction House with a higher asking price (Blizzard Entertainment 2009b).

Professions	*Gathering Skills*	*Secondary Skills*
Alchemy	Herbalism	Cooking
Blacksmithing	Mining	First Aid
Enchanting	Skinning	Fishing
Engineering		
Leatherworking		
Tailoring		
Jewelcrafting		
Inscription		

Table 1. A list of professions in World of Warcraft, sorted by type.

Some of the above professions are more costly than others. For example, much of what is needed for an alchemist can be obtained by choosing herbalism and fishing. Engineers can create many of the components they need for their craft via mining, but they also require a number of rare components from leatherworking, herbalism, and tailoring (Blizzard Entertainment 2009b). In this case and compared to an alchemist, an engineer may rely more on purchasing materials at the Auction House, or their own alternate characters, to craft items. As such, engineering ends up being a very expensive trade.

The cost of being a gatherer is really only the time it takes to traverse the virtual world of Azeroth harvesting materials. Areas of the world map are divided into zones that support character leveling; quests and monsters in each zone are intended for characters at specific stages of their in-game progression. Subsequently, since characters are meant to choose their professions early on, the materials available in each zone are also level-appropriate.

For example, copper ore is the first type of ore a miner will encounter. Most copper is dispersed in the mountainous areas of the lower level zones in the virtual world. As the player mines more and more copper ore, their mining skill will increase. At the same time, the player is also expected to be completing quests and killing monsters to gain experience points. Once the level of the player has become too high level to continue progressing in their current zone, they are directed, usually via a quest, to travel to the next zone in the game. Soon, the miner will encounter more tin ore deposits than copper. In order to be able to harvest this material, their skill in mining will need to have reached a certain level, currently 65 (Blizzard Entertainment 2009b). Players who do not wish to spend time looking for the materials themselves may choose to purchase them from another player at the Auction House instead.

When Blizzard added inscription to the list of professions, a number of players who leveled this trade skill feverishly profited greatly in-game. Those players who took inscription as their profession were able to create special items called glyphs that other players could buy and use to enhance their character's damage output, spell power, etc. Based on our personal observations, eventually the number of high level scribes grew and the quantity of glyphs on the market increased, resulting in a gradual price reduction.

While familiarity with in-game progression is helpful in mastering professions, it is important to note that the creation of these virtual items requires no real-world skills. The items crafted by players (weapons, armour, potions, etc.) are all generated using familiar 2D interaction metaphors, such as selecting items from menus and lists (see Figure 2). In this figure, the character is a tailor and knows several 'recipes' to create cloth items. Selecting a recipe from the list displays what ingredients or reagents are required to make the item. If the character has the necessary items, they simply click create, and the item is created. In Figure 2, the tailor has selected the recipe for *Ghostweave Pants*. The tailor has enough ingredients to make two pairs of these pants (as indicated by the number in square brackets beside the recipe name). Any resultant 3D content, such as a piece of armor, which is rendered on the character when worn, has actually been created in advance by skilled 3D modelers at Blizzard. These pieces of equipment are not customizable with regard to appearance or fit – the design of the end product has been pre-produced elsewhere.

Figure 2: A tailor's menu in World of Warcraft.

While the economics of character professions and the Auction House are built-in to *World of Warcraft,* there are a number of emergent economic exchanges that are not part of the game engine. Within this category of emergent exchanges, we can consider two distinctions: those that occur within the virtual world, and those that involve exchanges that cross the barrier between real life and the game world. Examples of the latter would be sites that allow players to purchase gold in exchange for real world currencies, and real world auction sites like eBay that have been used to auction off virtual goods (Campbell 2008).

The internal category of emergent economic exchanges is really quite fascinating. Players have been known to pay gold to other players for a variety of in-game services outside of those which are part of the game engine. For example, high level mages have the ability to create portals between major cities in the game world. A player may offer a mage gold in exchange for the ability to access a portal to another part of *Azeroth*. Similarly, warlocks have a summon spell that can instantly teleport any willing player character to their current location (Blizzard Entertainment 2009b). Players have also been known to call upon warlocks for this service. How much a player charges or offers to pay for access to portals or summoning can vary greatly, but the fact remains that the aforementioned examples represent the New-Style Services outlined in Noam's article (Noam 2007).

There are other examples of these New-Style Services that fall under the internal category of emergent economic exchanges in *World of Warcraft.* For example, players are often willing to pay higher level characters to 'run them through' difficult dungeons or quests. Many of these dungeons and quests require groups of players to complete them, and usually take a lot of time as the enemies are quite difficult to kill. A higher level player character can usually do much more damage than is required to kill these enemies so that including one in your group means that you can finish the dungeon or quest quickly and with fewer fatalities.

Comparatively, *Second Life* is a very different virtual environment both in its general use and economic systems. For example, while the trade of real world currencies for virtual currency is frowned upon by Blizzard, this feature is built-in to *Second Life*'s client program, the *Second Life Viewer* (Linden Research 2009a). Citizens of *Second Life* can currently purchase \$1000 Linden dollars for approximately \$4.15 USD.[i] Members of *Second Life* can easily attach a credit card to their account so that they can easily purchase Linden Dollars at the click of a button via the *Second Life Viewer*. There are also a number of resellers of Linden Dollars who offer conversion via other currencies.

The fundamental techniques required to create 3D content in *Second Life* are somewhat similar to those used in professional 3D modeling software packages, such as Autodesk's *Maya* (Rymaszewski et al. 2007). Residents can create objects within the virtual environment through the combination and manipulation of a limited number of geometric primitives or *prims*. These include: cubes, prisms, pyramids, tetrahedrons, cylinders, hemi-cylinders, cones, hemi-cones, spheres, hemispheres, torus, tubes, rings, trees, and grass. Figure 3 shows the top portion of the *Build* window, which residents use to create and manipulate objects in world.

Once a primitive shape has been created, residents use the most basic 3D manipulation techniques to position, resize, and reshape these objects. Figure 4 illustrates the three main techniques: position, rotation, and stretch. While a prim is being manipulated, a set of coloured handles appears around the object. The shape and position of these handles depends on which manipulation technique has been selected by the user. The colour of these handles correspond with 'real-world' directions within *Second Life*: red for east/west (the *x* axis), green for north/south (the *y* axis), and blue for up/down (the *z* axis) (Rymaszewski et al. 2007). Prims can also be positioned using coordinates within *Second Life*.

There are a number of general object properties, separate from its shape that can also be set by the user. These are all accessed via the general tab of the *Build* window. These properties include, but are not limited to, the object's name, description, creator, owner, and ability to be shared, copied, or modified (Rymaszewski et al. 2007). Whether or not an object may be modified, copied, or transferred is significant. It is not uncommon for sellers to set these properties such that once their virtual product has been purchased, the buyer is unable to change, create copies, or share it. As such, any other residents interested in acquiring the same virtual product would have to buy their own.

☐ Keep selected
☐ Copy selected
☑ Center
☐ Rotate
Selected objects:
primitives:

○ Position
○ Rotate (Ctrl)
● Stretch (Ctrl-Shift)
○ Select Texture
☐ Edit linked parts
☐ Stretch Both Sides
☑ Stretch Textures
☑ Use Grid
Ruler: World
Selected objects: 1
primitives: 1

Figure 3: The build window in Second Life.

The skill set required for 3D content creation in *Second Life* is not limited to 3D modeling skills: residents may also use external programs, such as *Adobe Photoshop,* to create textures and skins for their objects. Or they can use 3D modeling software to create novel forms of 3D geometry, such as a statue. Also, if the object has any resultant interactivity, residents will have to learn the *Second Life Scripting Language*. For example, once a resident constructs a chair in *Second Life*, they will then want to script the chair so that avatars will be able to 'sit' in the chair. A number of third-party development tools designed specifically for *Second Life*, such as *PrimDocker*, *EasyTexture*, and *TexturePallet*, have emerged in order to make 3D content creation and texturing easier for residents. A number of tutorials and in-world content developer training groups also exist.

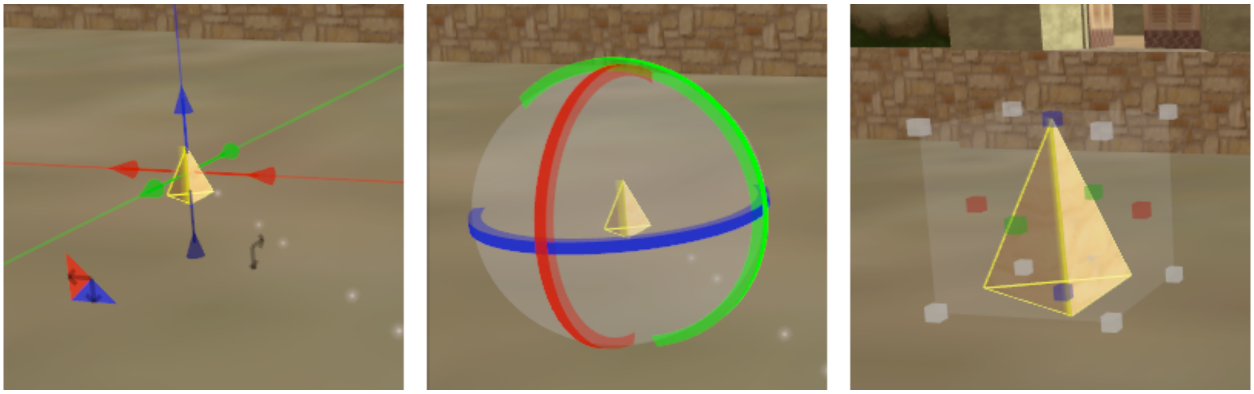

Figure 4: Three 3D manipulation techniques in Second Life: position, rotation, and stretch.

Based on experience, comments by players, and research results discussed earlier, we believe that the learning curve for content creation in *Second Life* is quite high and directly impacts supply and demand in-world. While there is no true 3D content creation in *World of Warcraft,* players can create virtual products (weapons, armour, etc.) with much greater ease. Consider content creation in these two virtual worlds as being on a scale: creating objects in *Second Life* is very difficult while *World of Warcraft* is very easy.

The aforementioned third-party tools and training groups are beneficial to users. However, the fact that so many exist is testament to the high learning curve of *Second Life*'s internal content creation interface. Adding third-party tools is a 'band-aid solution' and users of the software would be better served by an improved interface. Improving content creation in *Second Life* narrows the gap between it and *WoW*.

The figure below illustrates an approximation of the supply and demand model described above. If supply is low and price is high, demand is also high. If supply is high, and price is low, demand is also low. We hypothesize that supply is impacted by content creation interfaces in these environments. Just as a low number of glyphs on the market set the prices of these items initially high, a low number of high quality items in *Second Life* can result in high prices. If the market is flooded with high quality virtual products, then demand is low and prices will likely have to be reduced.

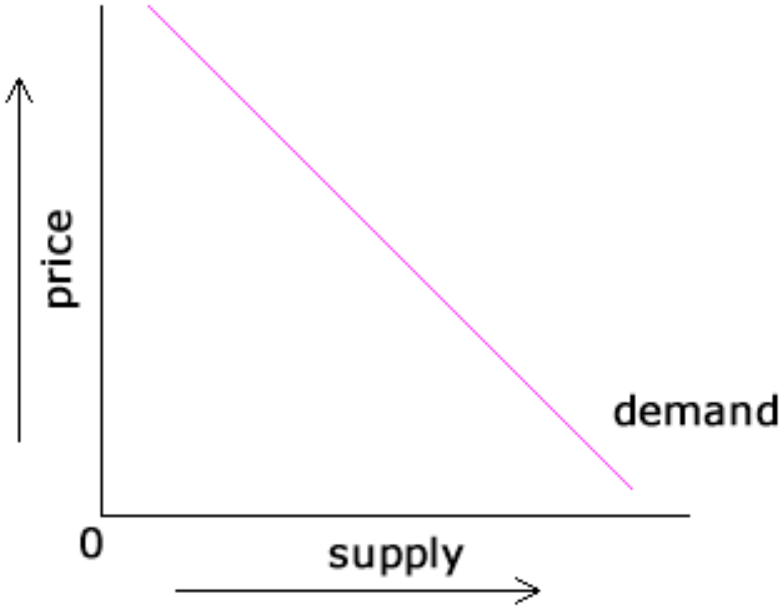

Figure 5: Approximation of supply and demand model.

In the following section, we propose guidelines for 3D user interfaces and make suggestions for 3D content creation systems in virtual worlds. We look to what previous work has demonstrated with regard to 3D user interfaces for 3D content creation, including some of the fundamental problems in 3D object selection and manipulation. We suggest that virtual worlds that follow such guidelines would benefit from a greater number of users creating virtual content. Users could generate their own content more easily, reducing reliance on a relatively small number of skilled users. However, this may not be too disruptive for skilled users, as they too could potentially enjoy higher levels of productivity using these improved user interfaces.

3. Solutions and recommendations: Guidelines for 3D user interfaces

We present a set of guidelines based on observations from previous work (Boritz and Booth 1997, 1998; Oh and Stuerzlinger 2005; Poupyrev et al. 1998; Smith et al. 2001; Teather et al.

2009a; Teather et al. 2009b; Teather and Stuerzlinger 2008b) as well as recent research in visual perception (Obayashi 2001). The intent of these guidelines is to help designers of games and virtual environments to develop intuitive 3D manipulation techniques. Most of these guidelines were developed specifically with the intent of improving 3D scene creation systems, i.e., positioning objects in a 3D environment. Many also generalize to object-level editing.

We work under the assumption that a typical 3D manipulation task can be decomposed into the following three distinct phases:

1. The selection phase, during which the user indicates which object they intend to manipulate.
2. A positioning phase, where the selected object is brought into the vicinity of the target area.
3. A 'fine-tuning' phase, where the object is rotated and positioned relative to the target.

The distinction between the first and second phase is the same as in Bowman et al.'s taxonomy (Bowman et al. 1999). The third phase is based on the observation that few people, if any, rotate and move the object simultaneously. While experts may rotate and translate an object simultaneously, this is something that novices do not appear to do. We do not believe that further decomposition of these manipulation phases is warranted, at least for novice users. We propose that the entire act of positioning an object be handled at once, without requiring the user to think in terms of movement along each of the three separate axes. As 3D rotations introduce a whole new layer of complexity to the problem, we limit ourselves to the 3DOF task of positioning objects in 3D in the scope of this article. Note, however, that many of the guidelines presented below may also be extended to rotation tasks.

We now introduce a set of guidelines for designing 3D object movement techniques. These guidelines encapsulate what we consider to be the most important design decisions for 3D object movement techniques.

1. Avoid floating objects.

In the real world, (almost) all objects are attached or connected to other objects. Floating objects are exceptional and our experimental observations suggest that most novice users are surprised when an object starts to float when moved. That indicates that the correct default for any 3D object movement technique is that objects should stay in contact with the rest of the world! However, most 3D modeling/CAD systems allow objects to 'float' in space by default, which we see as an area ripe for improvement. Solutions to this problem include gravity, constraints, or contact detection to always keep objects in contact with others, as well as other similar techniques. Similarly, when performing 3D mesh-level edits, vertices that comprise the mesh should always stay connected.

2. Objects should not interpenetrate each other.

Many novice users are confused when objects interpenetrate each other because it is difficult to tell which components belong to what object. Complex meshes with self-intersecting parts exacerbate this problem. Usually, novice users cannot easily resolve such problems. Incorporating collision detection/avoidance into movement techniques solves this problem. Today, the necessary computations are easily performed in real-time, even for complex scenes (Govindaraju et al. 2003; Knott and Pai 2003). Note that there may also be certain situations where relaxing this guideline may be beneficial. As an example, attempting to insert a peg into a tight hole may actually be easier if the objects can pass through one another. However, in general, major intersections should not occur.

3. Support relative positioning of objects by bringing them in contact with one another.
The paradigm of sliding an object on the surface of another until it reaches the desired position is a very natural way to position objects. This is easily demonstrated by watching a child position toy blocks. To implement this in a computer system, one must choose a movement surface from the set of surfaces of the static scene and then displace the moving object relative to that surface. One good way to realize this is by using constraints on object movement, see above. Another option is to ensure that objects always remain in contact with the rest of the scene. This also applies to object mesh editing. When moving vertices (or groups of vertices) of the mesh, it may be beneficial for them to remain 'on' the surface by default, while still allowing for movement orthogonal to the surface with another, separate user interface mechanism.

4. Only visible objects can be manipulated.
Users typically do not even try to manipulate objects that are not visible. Instead, they tend to rotate or move the viewpoint so that the desired object becomes visible. One indication for this is that previous work found that the most efficient techniques are based on the notion of ray casting (Grossman and Balakrishnan 2006; Poupyrev et al. 1998; Ware and Lowther 1997) or occlusion (Bowman et al. 1999). Ray casting identifies the first object that is visible along an infinite ray from the manipulation device into the scene. Occlusion is similar, except that it involves the user blocking the object to be selected with their hand, or another object. Hence, we suggest that it is sufficient to allow the user to select all objects from a 2D image (Ware and Lowther 1997), rather than using full 3D cursor selection techniques. And indeed, researchers argue that all ray casting techniques can be approximated as 2D techniques (Poupyrev et al. 1998).

5. The most important cues for judging 3D position in real scenes are perspective and occlusion.
As documented by research into visual perception, people judge 3D position based on several depth cues. Besides perspective, the most important cue for 3D position is occlusion (Wickens and Hollands 1999), i.e., closer objects visually block farther objects. In our previous work (Oh and Stuerzlinger 2005; Teather and Stuerzlinger 2008a; Teather and Stuerzlinger 2008b), we used a system that supported only the perspective and occlusion depth cues, while allowing users to easily move their viewpoint. The results from these studies indicated that this combination of cues is sufficient for humans to understand the relative 3D positions of objects, even in the absence of 3D stereo vision, as long as navigation is quickly accessible. Finally, it is interesting to note that other research confirmed that from an end-user's point of view, most stereo technologies are not very mature and are tiresome and/or problematic if used frequently (Diner and Fender 1993). In other words, the addition of stereo viewing to a system does not appear to greatly increase the usability of the system.

6. Avoid technical computer graphics techniques such as 'handles' and '3 orthogonal views'.
Using handles or widgets to move an object in 3D are both instances of indirect manipulation techniques. In the domain of (2D) desktop environments this idea was very rapidly eclipsed by the idea of direct manipulation (Shneiderman 1987), as this paradigm proved to be much simpler to understand. Furthermore, it has been shown that novice users can manipulate 3D objects more

effectively in a single perspective view and without handles when intelligent manipulation techniques are used (Oh and Stuerzlinger 2005). This also generalizes to vertex-level operations, which should be performed relative to the object the vertices belong to. To support higher precision requirements, systems can support direct coordinate entry using the keyboard for object placement. However, this should not be the default manipulation technique!

7. In general, 3DOF or 6DOF input devices provide less precision than 2DOF input devices.
A human hand held in free space will jitter more than a hand that is supported by a physical surface. Consequently, input devices that are physically limited to 2DOF tend to be more precise and hence usually afford also more efficient manipulation. In virtual and augmented reality research, this has been already realized through the adoption of techniques that involve the addition of a physical supporting surface, such as the Personal Interaction Panel (Szalavári and Gervautz 1997), or physical props (Lindeman et al. 1999a; Lindeman et al. 1999b). Such techniques effectively transform a 6DOF input device into a physical 2DOF input device.

8. Use the entire area of visual overlap of the moving object with the static background scene when deciding the position of the object.
Practically all techniques for 3D object motion use only the current position of the cursor to compute the 3D position of a moving object. This effectively reduces the computation to a point mapping problem. However, research into vision in primates discovered that the perceptive field for an object that is being held in the hand covers the whole object (Obayashi 2001). In other words, there is strong evidence that the whole visual area of an object is used to judge 3D position. And indeed, previous studies have shown that area-based techniques work better than point-based techniques (Oh and Stuerzlinger 2005).

4. Discussion and conclusions

Virtual worlds are not limited to the realm of ‘gaming’ – corporations and educational institutions have taken advantage of environments like *Second Life* for a variety of reasons. Unfortunately, the learning curve presented by the interface of many 3D virtual worlds has burdened these organizations with the added complexity of having to learn how to use the virtual world’s interface. Novices and non-skilled users may be unable to generate their own content, forcing them to purchase it from more skilled users. By increasing the usability of 3D content creation systems, virtual worlds would enjoy a greater degree of productivity from both existing content producers, as well as new novice producers who were previously discouraged by the difficulty of existing user interfaces for 3D content creation and manipulation. We speculate that this, in turn, might increase the population within these environments, as users begin to try the environment to experience the freedom of effectively creating their own experience.

Specifically, we predict that user experience would be improved in four key areas. Firstly, virtual worlds that follow these guidelines would benefit from a greater number of users creating virtual content. In the case of virtual worlds that are built on the ‘metaverse’ paradigm, such as *Second Life,* novice users may feel as though they have little to contribute, which is a factor that can lead to them abandoning the virtual world. The second and third areas of potential improvement are user retention and benefits to novices. When considering a 3D content creation system with a high learning curve, it is conceivable that some users have decided not to generate their own content because the system was too difficult to learn. If this has an effect on user retention, a system that is more intuitive might engage users who would have otherwise been too discouraged. Similarly, a system that is easier to use would be more inviting to novice users, possibly encouraging them to generate their own content much sooner. Lastly, a solid reputation for good usability in content creation and modification will also attract more new users to any system.

The proposed guidelines outlined within this article would also affect Noam's virtual world business models (Noam 2007), specifically V-Commerce. Supply and demand in virtual worlds is directly related to the ease with which users can generate their own content to sell. In one respect, introducing more intuitive 3D user interfaces could potentially disrupt such an economy, flooding the market with user generated content. Conversely, new users would be able to more quickly focus on their initial reasons for joining the virtual world, rather than wasting time learning how to *use* it.

While this discussion has been framed with virtual world economics in mind, the supply and demand model is an interesting byproduct of a more pressing issue: the design of 3D user interfaces for content creation. The aforementioned guidelines apply to 3D content creation in any context. Virtual worlds that have been designed for other purposes, such as education, would also benefit from these guidelines as well. Future developments in virtual worlds and interfaces for 3D content creation could learn a lot from the success Electronic Arts' *Spore*, in which users are treated like sculptors rather than mathematicians.

5. References

Areanet (2009), *Guild Wars*, Seoul, Korea: NCSoft

Autodesk (2010). *3D Studio Max*, Montreal, QC: Autodesk.

—— (2010). *Maya*, Montreal, QC: Autodesk.

Bier, E. (1987), 'Skitters and jacks: interactive 3D positioning tools', in Proceedings of the 1986 workshop on Interactive 3D graphics. Chapel Hill, North Carolina, United States: ACM, pp. 183-196.

Blizzard Entertainment Inc (2009a), 'Blizzard Store', http://www.blizzard.com/store/. Accessed 9 September 2009.

Blizzard Entertainment Inc (2009b), *World of Warcraft* Irvine, CA:Blizzard Entertainment Inc.

Boritz, J. and Booth, K. S. (1997), 'A study of interactive 3D point location in a computer simulated virtual environment', in Proceedings of the ACM Symposium on Virtual Reality Software and Technology - VRST '97. Lausanne, Switzerland: ACM, pp. 181-187.

Boritz, J. and Booth, K. S. (1998), 'A Study of Interactive 6 DOF Docking in a Computerized Virtual Environment', in Proceedings of the Virtual Reality Annual International Symposium: IEEE Computer Society: IEEE, pp. 139 - 147.

Bowman, D. A, Johnson, D. B., and Hodges, L. F. (1999), 'Testbed evaluation of virtual environment interaction techniques', in Proceedings of the ACM symposium on Virtual reality software and technology. London, United Kingdom: ACM, pp. 26-33.

Bowman, D. A. , Kruijff, E. LaViola, J. and Poupyrev, I. (2004), *3D User Interfaces: Theory and Practice*, Boston : Addison-Wesley.

Brandon, J. (2007), 'The top 8 Second Life virtual businesses', http://www.pcadvisor.co.uk/news/index.cfm?newsid=9279. Accessed 21 May 2009.

Campbell, D. (2008), 'Virtual Economics', *Region Focus*, 12:1, pp. 18-22.

Conner, B. D., Snibbe, S., Herndon, K. P, Robbins, D. C., Zeleznik, R. C. and van Dam, A. (1992), 'Three-dimensional widgets', in Proceedings of the 1992 symposium on Interactive 3D graphics. Cambridge, Massachusetts, United States: ACM, pp. 183-188.

Diner, D. B. and Fender, D. H. (1993), *Human Engineering in Stereoscopic Viewing Devices*: New York: Springer.

Ducheneaut, N., Ming-Hui W., Yee, N. and Wadley, G. (2009), 'Body and mind: a study of avatar personalization in three virtual worlds', in Proceedings of the SIGCHI conference on Human factors in computing systems. Boston, MA, USA: New York: ACM, pp. 1151-1160.

Google (2007), *Sketchup*, Mountainview, CA: Google.

Govindaraju, N. K., Redon, S., Lin, M. and Manocha, D. (2003), 'CULLIDE: interactive collision detection between complex models in large environments using graphics hardware', in Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. San Diego, California: Eurographics Association, pp. 25-32.

Grossman, T. and Balakrishnan, R. (2006), 'The design and evaluation of selection techniques for 3D volumetric displays', in Proceedings of the 19th annual ACM symposium on User interface software and technology. Montreux, Switzerland: ACM, pp. 3-12.

Jagex Ltd. (2009), *RuneScape*, Cambridge, UK: Jagex Ltd.

Kalning, K. (2007), 'If Second Life isn't a game, what is it?' http://www.msnbc.msn.com/id/17538999/. Accessed 1 April 2010.

Knott, D. and Pai, D. K. (2003), 'CInDeR: Collision and Interference Detection in Real-time using Graphics Hardware', in Proceedings of Graphics Interface 2003, Canadian Information Processing Society, pp. 73 - 80.

Lindeman, R. W., Sibert, J. L. and Hahn, J. K. (1999a), 'Hand-held windows: towards effective 2D interaction in immersive virtual environments', in Proceedings of Virtual Reality, 1999. IEEE, pp. 205-212.

—— (1999b), 'Towards usable VR: an empirical study of user interfaces for immersive virtual environments', in Proceedings of the SIGCHI conference on Human factors in computing systems. Pittsburgh, Pennsylvania, United States: ACM, pp. 64-71.

Linden Research Inc. (2009a), 'Second Life - Frequently Asked Questions', http://secondlife.com/whatis/faq.php. Accessed 1 September 2009.

—— (2009b), *Second Life*, San Francisco, CA: Linden Lab.

Maxis (2008), "Spore", Redwood City, CA: Electronic Arts.

McArthur, V. (2008), 'Real ethics in a virtual world', in Proceedings of the SIGCHI conference on Human factors in computing systems. Florence, Italy: New York: ACM, pp. 3315-3320.

Noam, E. M. (2007), 'The Dismal Economics of Virtual Worlds', *The DATA BASE for Advances in Information Systems*, 38:4, pp. 106 - 109.

Obayashi, S., Suhara, T., Kawabe, K., Okauchi, M. J., Akine, Y., Onoe, H., and Iriki, A. (2001), 'Functional brain mapping of monkey tool use', *Neuroimage,* 14:4, pp. 853-861.

Ocampo, J. (2008), 'Spore Creature Creator Hands-On', http://pc.ign.com/articles/875/875963p1.html. Accessed 30 April 2010.

Oh, J. and Stuerzlinger, W. (2005), 'Moving objects with 2D input devices in CAD systems and Desktop Virtual Environments', in Proceedings of Graphics Interface 2005. Victoria, British Columbia: Canadian Human-Computer Communications Society, pp. 195-202.

Poupyrev, I., Ichikawa T., Weghorst, S. and Billinghurst, M. (1998), 'Egocentric Object Manipulation in Virtual Environments: Empirical Evaluation of Interaction Techniques', 'n Proceedings of Eurographics'98, the 19th annual Conference of the European Association for Computer Graphics, pp. 41-52.

Rymaszewski, M., Au W. J., Wallace, M., Winters, C., Ondrejka, C. and Batstone-Cunningham, B. (2007), *Second Life: The Official Guide*, Indianapolis, Indiana: Wiley Publishing Inc.

Shneiderman, B. (1987), 'Direct manipulation: A step beyond programming languages', in R. M. Baecker (ed.), *Human-computer interaction: a multidisciplinary approach*, Morgan Kaufmann Publishers Inc., pp. 461-467.

Smith, G., Salzman, T. and Stuerzlinger, W. (2001), '3D scene manipulation with 2D devices and constraints', in Graphics interface 2001. Ottawa, Ontario, Canada: Canadian Information Processing Society, pp. 135-142.

Szalavári, Z. and Gervautz, M. (1997), 'The personal interaction panel - a two-handed interface for augmented reality', Computer Graphics Forum 16:3, pp. 335 - 346.

Teather, R. J. and Stuerzlinger, W.. (2008a), 'Guidelines for 3D Positioning Techniques', in Proceedings of FuturePlay '08, ACM, pp. 61-68.

—— (2008b), 'Assessing the Effects of Orientation and Device on (Constrained) 3D Movement Techniques', in IEEE Symposium on 3D User Interfaces 2008, IEEE, pp. 43-50.

Teather, R. J., Allison, R. S. and Stuerzlinger, W. (2009a), 'Evaluating visual/motor co-location in fish tank virtual reality', in IEEE Toronto International Conference - Human Factors and Ergonomics Symposium, IEEE, pp. 624-629.

Teather, R. J., Pavlovych, A., Stuerzlinger, W. and MacKenzie, I. S. (2009b), 'Effects of tracking technology, latency, and spatial jitter on object movement', in IEEE Symposium on 3D User Interfaces. Lafayette, Lousiana, USA: IEEE, pp. 43-50.

Ware, C. and Lowther, K. (1997), 'Selection using a one-eyed cursor in a fish tank VR environment', ACM Trans. Comput.-Hum. Interact. 4:4, pp. 309-322.

Wickens, C. and Hollands, J. (1999), 'Spatial Displays', *In Engineering Psychology and Human Performance (3rd Edition)*: Prentice Hall.

Contributor details

Victoria McArthur is a Ph.D. candidate in the Communication & Culture programme at York University, Canada. She received her honours BA in Music from Brock University in 2007 and her MA in Interdisciplinary Studies at York University in 2010. In 2003 she received a full scholarship to study at Nagoya Gakuin University in Japan, where she graduated with a certificate in Japanese area studies. Her research is currently funded by the Elia Scholars Program.

Contact: 3013 TEL Centre, 88 The Pond Road, York University, 4700 Keele Street, Toronto, Ontario, M3J 1P3
E-mail: vickymc@yorku.ca

Robert J. Teather is a Ph.D. candidate in Computer Science & Engineering at York University. He received his honours B.Sc. in Computer Science from Brock University in 2003 and his M.Sc. in Computer Science at York University in 2008. His primary area of research is comparing 2D and 3D user interfaces, and quantifying their differences. Robert’s research is funded by the Natural Sciences and Engineering Research Council of Canada.

Contact: Department of Computer Science and Engineering, York University, CSE 1003, 4700 Keele St., Toronto, Ontario, Canada, M3J 1P3
E-mail: rteather@cse.yorku.ca

Dr. Stuerzlinger graduated with a Doctorate in Computer Science from the Technical University in Vienna, Austria in 1993. Then he moved to the Johannes Kepler University of Linz, Austria. Supported by an Erwin-Schrödinger fellowship Dr. Stuerzlinger visited the Department of

Computer Science at the University of North Carolina in Chapel Hill in 1997 (hosted by Prof. F. Brooks). In 1998, Dr. Stuerzlinger was appointed to the Department of Computer Science at York University in Toronto, Canada. There, he is an associate professor and leads the Interactive Systems Research Group. Dr. Stuerzlinger has supervised more than 30 graduate students to completion and published more than 80 refereed scientific papers. His group currently consists of a postdoc and 10 graduate students.

Contact: Department of Computer Science and Engineering, York University, CSE 1003, 4700 Keele St., Toronto, Ontario, Canada, M3J 1P3
E-mail: wolfgang@cse.yorku.ca

---

[i] Exchange rate obtained from the Linden Exchange section of secondlife.com on September 24, 2009.