Scale and the construction of real-world models in Second Life

Dr Jonathan C. Roberts, Dr. Robert Gittins, Bangor University
Dr Rhys Thomas, Leeds University

Abstract

Virtual environments are often used for training. In particular, virtual worlds can be used for situation-awareness. Such models allow users to gain an understanding of a real-world location or walk-through scenarios that may incorporate many participants and achieve training to develop skills and competencies. One of the main challenges for the developer is to create virtual models that are appropriately sized in comparison with the avatars. In-world measurements are different to those in the real-world. In-world models have exaggerated avatar heights and often use third-person viewing projections. This article describes our experience with developing a scale model. We have used Second Life to provide a case study for understanding virtual design issues and challenges of scale in development. We discuss these challenges and describe our experiences of developing real-world models in virtual environments.

Keywords

Scale
virtual environments
realism
group and organization interfaces
web-based interaction
1. Introduction

Scale models have been used throughout history. Leonardo Da Vinci created scale models of his ideas and military commanders have used models or sketches to plan battles, while modellers throughout history have re-created both realistic models and functional representations that describe the operations of inventions. These surrogate visualizations have many uses and subsequently have multiple benefits.

Three-dimensional models, then, have long been used to represent the real world. Three-dimensional virtual environments allow multiple remote users to be co-located in the same virtual world and interact with the world through walking, flying and manipulating objects. Some of these virtual worlds mimic the real world while others are fantasy worlds. In particular, realistic models can be used to train new scenarios to users and allow participants to develop proficiencies or to hone their skills in preparation to achieving similar tasks in the real world (Stuck et al. 2009). However, although many researchers have created realistic virtual models of various domains and discussed their validity (Wergles and Muhar 2009), there has been little documented research of how to generate appropriately scaled models in specific virtual environments.

Our motivation is to create virtual environments that are both useful and suitable for situation-aware tasks, where users gain an understanding of a real-world environment by first experiencing a virtual world simulation. Situation awareness entails being aware of the situation, of what else is happening around, being aware of how various facets fit together, and a perception of how the user is part of that world and how their actions will impact on their goals and other people’s goals. Scaled environments are important for situation-aware tasks because it is necessary for users to gain a correct understanding of the world. For example, a scene should enable the user to perceive and estimate how long it would take to walk between two points or notice key features to enable them to navigate the corresponding real-world scene. As well as to address navigational tasks, the simulated world could be useful for line-of-sight tasks, allowing questions to be asked such as ‘can I see the person in that doorway if I stand here?’ This includes enabling users to interact with the simulated environment, together with other in-world users, in more or less complex activities that can be transferred to real-world settings.

This article focuses on the challenge of scale, where the width, height and depth of the objects in the drawing are proportional in every way to that of the original object. We highlight below where the challenges lie and discuss potential solutions to enable developers to create suitably scaled and realistic-looking virtual environments.

Throughout this article we use our modelling of the area known as ‘Golden Square’ in London in the Second Life online virtual environment (SL) (Linden Research 2003) as a case study. Figure 1 shows a screenshot from the finished model. The aim of the Golden Square project was to create an environment that would be usable and appropriate for users to perform situation-aware tasks. The work was achieved by the authors for our clients who wish to remain anonymous but wanted to explore the feasibility and potential
of developing realistic virtual environments to scale. Golden Square was chosen because it is a typical urban city environment. Significantly, it has a mix of man-made structures such as buildings and roads as well as a large garden area with trees and shrubs.

![Figure 1: A snapshot of our scaled virtual model of Golden Square in London that was developed in SL. The picture shows an avatar sitting on a bench who appears appropriately scaled with the environment.](image)

This article discusses some issues surrounding scaling and ways to achieve scaled environments, and it is structured as follows. First, we present some background information about the three main concepts: scale, virtual environments and real-world models. Second, we develop a ternary categorization of Production, Physical Environment and Perception (Figure 2). This categorization was developed through our creation of the Golden Square model. Finally, we discuss the presented ideas and include a short list of recommendations for developing scaled environments, which has been gathered from the literature and our combined experience.
Figure 2: We divide the Virtual Environment creation into three stages: Production, where the models are created and the choice of the infrastructure is made; Physical Environment, which includes the viewing environment (i.e., the software that renders the world) and the Devices that are used to both display the model and interact with it; and finally, Perception, where the user observes perceptual cues to calculate position and depth (et cetera), and to understand what the world represents.

2 Background and related work

In this section we discuss the three main concepts referred to in this article: virtual environments, scale and real-world models.

2.1 Virtual environments

SL has approximately eighteen million accounts registered, and is often the platform of choice for many tasks such as education (De Lucia et al. 2009) or medical training (Boulos et al. 2007). Created by Linden Lab in 2003, the source code for the SL Viewer is available to everyone. Developers can download the SL end-user software and make modifications and additions to the code. The SL client viewer enables users (‘Residents’) to interact with each other. Residents can navigate the world – built by the residents themselves – through walking, running, flying and teleporting or moving around in vehicles. They can socialize through joining groups, meeting people and speaking or typing, exchanging specific instant messages or document cards, and can buy and sell goods in Linden Dollars. Because of all this variety, SL provides a useful environment where scenes can be quickly modelled and explored.

There is evidence in the literature that supports the use of virtual environments for scientific research and to develop skills. Rosser et al. (2007) report on a strong correlation between those who demonstrated good computer game skills and good medical skills (in
this case laparoscopic skills), and many computer games have been used specifically for training purposes (Mitchell and Savill-Smith 2004). Gee (2008) argues that good games enable users to learn and stretch their minds. These games are both challenging and frustrating, but users can potentially transfer the learnt skills from that domain to another.

2.2 Scale and virtual environments

The term ‘scale’ has various meanings, but we use it in the sense derived from the Latin word *scala* or ladder. This acts as a tool with gradations to denote distances that can be used to measure something against that scale. We believe scale to be an important factor to model virtual worlds, especially for situation-aware tasks. Specifically, we wish to develop worlds that are scaled appropriately in terms of depth and height and that are suitably sized to the virtual avatar and natural for users to experience. We do acknowledge, however, that scale is only one attribute, and that for different tasks developers may need to spend time on other attributes to make the environment suitable for a particular task. Westerdahl et al. (2006: 164) write, ‘if [Virtual Reality] models are defined as models providing a full sense of presence, full-scale VR models are presumably not needed at all stages or for all tasks in the building process’. Sometimes scenes that precisely mimic the real world are difficult to navigate virtually (De Lucia 2009). While, on the one hand, navigating through many doors can generate a laborious and frustrating experience for its inhabitants, on the other hand, environments like SL offer additional types of navigation that are not found in the real world where users can teleport and fly to new locations, and thus engender new possibilities in communication and understanding of the virtual world (Hill and Lee 2009).

We wish to develop environments that are instantly usable. However, SL users can customize their appearance through changing several values where they can alter many aspects including body size and height. The height parameter changes several body features simultaneously. Users often create a ‘tall’ avatar without realizing that their avatar is taller than average, which has a direct implication on creating usable environments for those avatars. Users with smaller avatars – but who have a typical average height equivalent to a real-life height – may look ‘out of place’ in-world. Furthermore, it can be difficult for users to clearly understand the height of their own avatar during the customization process. One reason is that, in the avatar customization, height is a mix of several parameters. Height is given in a range 0–100 without explicitly correlating those numbers to an actual height (displaying the height information will be possibly rolled into later releases [Linden and Patton 2010]). This situation is further complicated by the functions in SL, such as the code function llGetAgentSize, which is used in many height measurement tools but produces inaccurate results. Consequently 0.17 needs to be added to get the actual sole-standing height. These are all issues that content creators in SL need to be aware of.

Avatar size affects the building process. To generate an appropriately scaled world in SL the in-world assets and avatars need to be sized relative to each other. Either a user needs to change the size of their avatar to view that world appropriately or the developer needs to size the world appropriately (Boulic et al. 2009). Third, input devices and control depend on avatar size. For example, Boulic et al. (2009) propose three strategies when using a full-body postural input: reference strategy, which controls directly a same-height
avatar scaled 1:1, visuocentric scaling strategy, which uses a height-differing avatar that is controlled through scaling the sensor input to match the height of the avatar, and egocentric scaling, which matches the avatar height to the user’s height.

2.3 Real-world models
‘Real-world models’ are computer-generated models that represent the real world. They are representations that stand in for the real environment. Online worlds such as SL are implicitly metaverses that copy principles from the real world. The physics of these metaverses are different to those in the real world, where users can perform actions that are not possible in the real world such as flying, or viewing themselves from a third-person viewpoint. Furthermore, the building materials do not mimic the real world and do not have real-world limitations, so developers can create buildings that defy the laws of gravity, for example. It is also possible to constrain users from operating some of these additional features (such as flying and teleporting) and include physics engines to simulate gravity or object properties to more closely mimic reality.

The goal of a developer may, in some cases, be to create the most accurate and realistic equivalent of the real world possible. These realistic worlds could contain accurate models, are displayed under natural-looking lighting conditions and demonstrate real-life physics (where virtual objects break or shatter naturally and collide and bounce off other objects correctly). However, such high-fidelity photorealistic environments are often impractical, difficult to create, require particularly skilled developers and are not usually interactive due to their high computational complexity. Although the representation of a virtual world may be ‘to scale’, it need not be realistic. There are two types of techniques that can be employed to provide the non-realisim: non-photorealistic techniques and abstract methods.

Non-photorealistic rendering methods depict the scenes in styles that are inspired from art. The renditions appear as hand-drawn sketches, line-drawings or cartoons. Users implicitly perceive that the information is less certain – which is indeed useful when representing information that has a high degree of uncertainty, such as virtual heritage (Roussou and Drettakis 2003). Moreover, users who are less familiar with computer visualizations may be more sceptical of photorealistic visualizations (Appleton and Lovett 2003), or the more realistic scenes may give the user a ‘wrong impression’ of the real scene. In particular, if the model is not 100 per cent accurate, the user may notice these minor mistakes and become frustrated, or they could unknowingly learn these nuances and expect them to be in the real scene.

While non-photorealistic methods offer alternative ways of viewing these worlds, other more abstract renditions are possible. Abstract depictions are formed by reducing the quantity of information present or changing the style of its presentation. They could encourage users to use their imagination more, and enable users to perform specific tasks faster. These additional views can enable the user to construct additional situation-awareness. Geographical maps enable users to view their location in comparison with other users (Boulos and Burden 2007). Bar charts or line graphs could be used to depict statistical information. These abstract depictions are achieved through emphasizing one attribute at the expense of others. Underground maps, for example, often display accurate
connections for each of the stations at the expense of their correct geographical locations. This means that it is often easy to navigate the underground railway itself, yet difficult to work out the relative position of those stations in real space. The task that the user has to perform should govern the type of visual depiction, and the developer needs to visually depict explicit links between the different views (Roberts 2007).

Both non-photorealistic and abstract renditions use symbols instead of realistic objects, which has several advantages. For example, with realistic scenes users may not be able to see through a tree; however, if a symbolic representation of a tree is included, then the user may be able to better understand the relative position of other objects. Such symbolism could be included as part of a standard toolkit for developers. But, it is often difficult for a developer to choose correct symbols or for the user to perceive them correctly (Orland et al. 2001). Whatever virtual world is created, it is made with signs and includes several conventions. Each world is created through images, words, symbols and various artefacts, thus participants need to learn the rules of the world to be able to interact with it. Many of these metaverses mimic the real world with similar icons and similar physics, so that users may take this semiology for granted; however, ‘[l]earning involves mastering, at some level, semiotic domains, and being able to participate, at some level, in the […] groups connected to them’ (Gee 2008: 49). It is thus important to balance the trade-off between realism and appropriateness, which depends on the specific task that the user is given.

The ‘realism’ versus ‘abstraction’ debate is well documented in the literature (Wergles and Muhar 2009). However, whether the environment is realistic, representational or abstractly depicted, when the information is being used to develop situational-awareness, there is a need to create models that are scaled appropriately. After the development of their real world Virtual Environment for Architecture and Urban Planning, Drettakis et al. (2007: 328) write that ‘the sense of scale given by the combination of realistic vegetation and human figures/crowds was an important effect which we had not suspected initially’. Although research has been done on the area of realism, little research has been done to provide practical approaches of how to make virtual environments that are based on real-world sites, that appear natural and are scaled appropriately in comparison with the size of an avatar.

Scaled environments provide specific challenges. To model these environments a developer needs to create models that enable users to make judgments on distances (see Section 6) – a problem that also occurs in real-world situations – and that allow users to compare the size of objects.

3. The categorization: production, physical environment and perception

To develop effective virtual models that are suitable for situation-aware tasks, we argue that scale should be considered throughout each stage of design and development. Various conceptual design processes have been followed by different developers and certainly many of these provide useful and practical solutions. Although some authors have acknowledged the lack of suitable design methodologies for virtual environment
development (Mansouri et al. 2009), we do not consider that scale has been thoroughly taken into account in those that do exist. For instance, (1) Kirner and Martins (1999) use an adapted software development process including the phases of Requirements, Design, Implementation and Evaluation Cycle. They only include a passing mention to scale: ‘a common scale was adopted for modelling all objects; the colours, forms and textures applied to the objects try to reproduce the original ones’. (2) Wilson et al. (2002) propose the Virtual Environment Development Structure that considers preparation, analysis, specification, building, implementation and evaluation, but make few comments on the challenge of scale. (3) Mansouri et al. (2009) describe the Web and Information Systems Engineering (VR-WISE) approach of specification, mapping and code generation with little mention of scale.

Based on our own experience and research, we propose that the components of a virtual environment can be divided into three parts we call ‘the three Ps’. This provides a convenient categorization which we will use to discuss issues of scale. To classify these parts we define them as: Production, Physical Environment and Perception (see Figure 2).

1. The **production** category includes any aspect that deals with the model itself. This includes design, creation, storing, layout and production of the scene. Models are usually developed using various geometrical shapes (the objects in the scene). Sometimes these models are hierarchical, while individual (or grouped) objects hold various attributes to determine appearance (colour, size, texture) and behaviour – the behaviours can be used to move the objects and give them autonomy in the environment. Properties such as weight, mass or whether the object is deformable can be included with the object, depending on the capabilities and functionality of the renderer. Although not the focus of this article, the development of the virtual-world infrastructure also fits in this category. A system developer can include extra functionality that would help modellers create effective scaled models, such as rulers, scales, markers or camera constraints.

2. The models then need to be realized into a **physical environment** for the user to view them. This physical environment consists of two parts: a renderer and the physical interface. Rendering takes ‘environment’ variables and ‘viewport’ information to generate the image that is displayed to the viewer. Different renderers can be used, from wireframe, photorealistic to non-photorealistic, and different parameters of the renderer can be changed to affect the quality of the displayed output or the speed of interaction. User interaction is included in this category as user navigation performs changes to the renderer. The user-interface may be achieved through different technologies such as desktop screen and mouse or immersive three-dimensional (stereoscopic) display devices on large screens.

3. Finally, the user interacts with the environment (the renderer and the model), uses cues in the environment (such as colours, shadows and shading) and perceives what the environment represents. They move their avatar to investigate and explore different parts of the model.
We present this categorization in the next three sections and describe our experience with the Golden Square project.

4. Production – modelling

Although the VR-WISE approach (Mansouri et al. 2009) is biased towards the creation of entire virtual worlds, it does provide a useful model for our ‘production’ category. Subsequently, (1) there are challenges with the preparation and gathering of the material. For instance, photographs need to be taken – of an appropriate size and resolution to be used for world development – and they may need to be edited and cropped. (2) There are various challenges to do with the overall scale and size of the virtual model built: will it fit into the size of the world? What is the appropriate height of a building in comparison to an avatar? How can the building be created to scale? (3) There are challenges to do with the creation and re-creation (or re-location) of the model being developed. Will the model fit into the world? Are there any modelling tools that may help with producing a scale model?

4.2 The production of Golden Square

During the production of the Golden Square model we followed this three-tier design, which we summarize as Preparation, Scaffolding Development and Main Development. In particular, various ‘scaffolding’ models were created – scaled correctly – that were used during the main building phase. Our scaffolding included scaled objects at specific locations and other markers that denoted the boundaries of the main components of the building. This meant that once the scaffolding was in place, the main building process would be easier.

4.2.1 Preparation

During the preparation phase of the Golden Square project, various materials were gathered. Photographs of building façades were taken and stored, along with photographs of objects (such as benches, waste bins, telephone boxes) and close-up textures of the objects. Aerial photographs and maps of the area were gathered. Each of the images used in the build were post-processed, in particular the images of buildings, because they were taken from street level, needed to be trimmed, de-warped and edited to remove obstructions so that they could be used as textures for the building phase.
Figure 3: The picture shows the Golden Square model being created. The ground plan can be clearly seen.

It is important to get the right photographs and materials before the actual building process begins. We tried three methods to gather pictures of the real-world scene: (1) video camera, (2) a fish-eye camera mounted on top of a car was driven round the square to create 360 degree moving images (3) still images were taken with a 12 Mega-Pixel single-lens reflex camera from street level. The video footage gave us a good overview of the whole area and was used as reference material to guide the build, but it did not end up being used in any of the final models. Likewise, the fish-eye footage helped us to gain an understanding of the whole set, but we found it was difficult to manipulate. The files were high quality and each frame needed to be warped from the fish-eye perspective. Most of the fish-eye footage was not used. A few frames, however, were used and merged with the still images in some places where the building façades were obscured. This was possible because the fish-eye camera was higher off the ground in comparison to the still images. In some places where it was impossible to retrieve the texture from the SLR photographs, we copied a similar texture from another photograph. As the still images of the building façades were taken from ground level, we warped these textures in Photoshop™ (Knoll et al. 1990–2008) to correct their orientation and remove the distortion. Obviously, the higher the building, the more warping was required, which generated lower quality images for higher parts of the building. Originally, we were worried that this would be noticeable, but in fact it does not appear to be any different from the other textures. Importantly, there is no noticeable difference from street level in SL.

4.2.2 Scaffolding Development

Scaffolding Development consisted of four parts: layout development, ground plan creation, height plan creation and tool building.

1. Aerial photographs and maps were consulted to work out the overall scale and size of the building plan. Building a large city to scale would be practically impossible in SL. Furthermore, simple objects (cones) were placed at the main corners of the build to mark the bounds of the build.

2. The ground plan was used as a reference during the entire project and was derived from high quality survey maps of the area. It was created by placing a large image on the floor. A .tga file of the square was imported, that was taken from a high quality map of the region, which had been simplified using a colour threshold and edge-detection algorithm. To place the plan into the SL environment it was applied to a prim as a texture. A large square 10x10m prim was laid on site and the .tga file was applied as a texture across this selection of prims using the EasyTexture tool (Tone 2010). An initial estimate of the scale was determined by the use of an in-world object. In the first instance a car was used. The size of a real-world car was determined in comparison with a human, which was translated into an in-world car that was then copied several times and parked bumper-to-bumper to confirm the width of the road on the ground plan. The final scale was made when the height plan was created.
3. The height plan (silhouette) was created by a reference building that sat directly on top of the ground plan. This was done to facilitate the orientation, scaling and construction of the Square. It was achieved by placing a flat prim as a building façade that matched the width of the building in the plan, and it was textured with an image of the building at that location. The height of the building was adjusted to correct the aspect ratio using an avatar that was positioned in front of the building. This scene was directly compared with a photograph of a real person standing at the same real-world location. Subsequently, the size of the reference building was adjusted until the avatar and the real person appeared to be the same size in respect of the related building. The size of the ground plan was then adjusted so that an acceptable fit was achieved for the new building. Various other landmarks were used to check the scaling, including lamp posts (see Figure 4). But we found that doorways in particular provided a good indication of scaling.

4. Several measuring instruments were created and used during the build. These included ‘markers’ as placeholders to identify positions of potential objects and three-dimensional rulers (Figure 5) to enable objects to be created in specific sizes. We used a grid to help build the objects to scale. This enabled the developer to quickly create an object in SL to a desired size (Figure 6). Furthermore, the Prim Docker tool (Reddebohm 2010) was used to align the objects.

Figure 4: The left image shows a photograph of a building façade and lamp post taken from Golden Square. The right image shows a similar view from our model in SL.
4.2.3 Main development
During the main development phase the silhouette of flat (non-textured) objects were enhanced with textured objects. The textures were first prepared as described in the preparation phase. Specifically the perspective crop tool in Photoshop™ was used to extract regions that were rectangular in reality and correct in perspective. This generated rectangular images of regions of the building. In addition, any occluding objects were also removed. This was achieved by copying textures from the same image or from other
images. Several regions of the same building were then combined to create a single large flat texture for the front face of the building. Where an area had no suitable texture, a copy of a similar region was put in its place. This texture was then applied to the flat prims of the building face using the EasyTexture tool. The appearance of each building was then either compared to a photo of the real building with a person standing in front of it or to the reference building where appropriate to determine whether the height of the building was correct. Adjustments to the height were made if necessary.

Once all of the buildings were created as flat surfaces, each was re-visited to assess whether any detail could be added to give the impression of depth. We decided that features such as recessed windows and doorways and protruding pillars were important. The prim that included the ground floor of the building was split into several parts which could be moved independently to give the impression of depth. Textures were then re-applied, and new ones created for surfaces that had not existed previously. Several buildings in the square had windows projecting beyond the main body of the building. Oriel windows were modelled using a cuboid for the centre and two triangular prisms for the sides. Bow windows were modelled using a hemi-cylinder. Textures were extracted from the photos and applied to these features. We repeated the textures over the entire face if individual textures were the same or unavailable. Subsequently, all the remaining ground-level buildings with common textures were likewise completed.

To enhance the realism of the buildings some other external features were created. Several buildings had steps leading up to the doors, which were modelled using cuboids and textured with appropriate concrete textures. Railings were added to the scene. The railings comprised of a base, composed of a hemi-cylinder on top of a cuboid, and the metal railings were modelled as a large thin area to which a railing texture with transparency was applied. Where present, steps down to a basement floor were modelled by a standard gravel texture because of a lack of sufficient photographic material to recreate these features exactly.

The Golden Square has several roads leading off the main square. Thus the final part of the scene building was to model these roads. It was impractical to model the roads in the same way. Therefore we used photographs to mimic these vanishing points. Several photos looking down the streets between buildings were edited in Photoshop to remove the portions of the buildings that were already modelled. These vanishing photos were then included in the world recessed from the building line to provide a better vanishing appearance. Finally, textures were created to fill the gaps between the buildings and the vanishing photographs.

5. Physical environment
There are several challenges with scale and the virtual physical environment that the user explores. In this context we refer to the physical environment to include the renderer/viewer to display the result and the physical devices used to interact with the environment.
5.1 Physical environment – challenges with the viewport

Virtual environments provide various viewing solutions. In SL there are two possible viewing positions. ‘follow camera’ (Figure 6) provides a third-person view with the camera approximately 2.5 m behind the avatar, in contrast to the ‘mouse look’ (Figure 7), which is a view that places the camera in the eyes of the avatar. Both views provide somewhat unnatural perspectives for the user where the field of view is narrower than in real life and without peripheral vision. This is an important concern for situation-aware tasks such as getting an overview of the area and navigation through the model.

The ‘follow camera’ is the default view in SL and places the user behind their avatar. It has several advantages. In particular, the user can view interactions between their avatar, other avatars and artefacts in the world. This affords an effortless interaction methodology where the user can readily click on their avatar to alter its appearance or get the avatar to pick something up. This mode is beneficial when in an open space, but not so convenient when the viewer enters classical scaled buildings and more confined spaces. The Golden Square example works well in ‘follow camera’ because the distances between building structures and other artefacts are quite long and a user can gain a good overview of the scene. We feel that the lack of peripheral vision is somewhat compensated by the camera being behind the avatar. However, one of the problems is that the camera can disappear behind a wall or into the structure, thus occluding the user’s view of their avatar. Furthermore, avatar customization is an important part of SL and subsequently the avatar may act to distract the user from their task rather than helping them. For more functional purposes such as meetings, exhibitions and events it is better to create landscapes and buildings with open spaces (see Hill and Lee 2009).

There are several advantages to using the ‘mouse look’: users control the viewport directly, objects directly in front of them are not obscured by their own avatar and they are not distracted by the avatar’s appearance or animation. However, when using fully immersed head-mounted-display devices Salamin et al. (2008) discuss that users who see their own avatar have increased proprioception – the ability to understand the movement and orientation of our body – when they use a third-person viewpoint. They write:

We could then say that walking action and distances evaluation without stereo vision is easier to do with the third-person perspective while target actions or hand manipulations such as opening a door can be better performed at the first-person perspective. (Salamin et al. 2006)
5.2 Physical environment – challenges with ways to move

Interaction devices and ways to move around the three-dimensional environment provide further challenges in a scaled environment.

Using the arrow keys or mouse operations to control the direction and speed of a walking avatar is the traditional way of moving in SL. In the Golden Square example we wished users to walk, such that they may act as they would in a real-world situation. Researchers have demonstrated that walking in place increases the user’s subjective view of presence of the world (Usoh et al. 1999). Our experience shows that users, especially those who are experienced, can become frustrated when constrained to walk in confined areas.

Flying offers a fast way to move around. It allows users to take direct routes, see distant objects and gain a holistic understanding of the scene. Users obtain a good understanding of the layout of the roads and how the city is formed and then drop down to gain a closer look. Flying can be used to gain location-awareness in the same way a geographic map is used. However, scaled buildings that mimic real-world constructions can cause problems. For instance, avatars are often unable to fly in buildings due to the ceilings and walls (De Lucia et al. 2009), and slow and fine adjustments of position are difficult to make.

Similarly, teleportation is a fast way to navigate large distances. In SL teleport points can be marked and saved in a personal inventory, searched for and collected from the search engine, received from friends who have collected them, or issued by organizations who would like you to attend their venue. Business contacts or friends can offer instant teleport invitations anywhere across the tens of thousands of regions in SL. The speed and
agility of movement in three-dimensional environments creates a mode of interaction beyond the classical real world experience, but teleportation causes the user to lose context information, and it is difficult for users to understand where they were in correlation to where they are now.

Although not very common, other methods of interfacing with the model need to be taken into account. Using real-world treadmill systems or walking in a room with motion sensors attached can provide a powerful interface to the virtual environment, but it usually requires additional hardware to be worn and to be situated in a special room with additional equipment. Thus, the technology generates some limiting factors. (1) The user cannot easily move to a new place. With a mouse, for instance, the user can pick up the mouse pointer and move it to start a new position, but this is difficult in a treadmill. (2) The room that the operator is located in is only a fixed size. The user will readily walk to the end of the room, yet they would not have walked the full distance of the virtual world.

Peck et al. (2009) compare various distortion techniques that enable users to physically walk in a restricted (tethered) location whilst believing (through the visual illusion) they are walking a long distance. Many of these techniques work by tricking the user that they are only moving or turning a short distance (through visual cues) while actually their body is turning a much larger distance. (3) The user can get physically tired through walking long distances. (4) This style of technology is currently very expensive, thus it would be prohibitively expensive to have multiple participants enacting in the world through the same type of interface.

Thus, as well as creating scaled landscapes and buildings, developers need to balance the usability of these worlds with the constraints of a given interface and the task that the user is required to perform.

6. Perception

There are certainly differences in the way people perceive scale in the virtual environment compared to the real world (Eggleston et al. 1996). Several researchers support the claim that making size judgments in virtual environments is less reliable than in the real world (Wuillemin et al. 2005) and that distances are often overestimated (Waller 1999).

Although useful, many of these evaluations are achieved on specific (and simplified) models. Functional environments have multiple objects that are placed in the near and far field of view. Scale should thus be considered in ‘context’ (Tufte 1997). Thus, provided the user has appropriate spatial cues, the perception of specific objects – that are known to the user – remains consistent wherever they are in the view. For example, users perceive a bus to be larger than a car wherever they are placed in the scene. Size-constancy is possible in complex virtual reality scenes (Luo et al. 2009). Our Golden Square model has known objects (lamp posts, cars, doorways etc.) that users could use to understand size and scale. Murgia and Sharkey (2009: 73) write
The accuracy of perceived distances appeared to be higher than in other studies where the participants were not provided with information on the relative size of the virtual objects used.

In monocular three-dimensional worlds appropriate depth cues such as shading and shadows are important factors to appropriate size perception. While stereovision can enhance size constancy and accommodation may not play a major role in monocular displays (Kenyon et al. 2008), depth cues play an important role in stereo three-dimensional displays (Watt et al. 2005).

In sum, the effectiveness of our perception of the virtual world and whether it appears scaled in proportion to an avatar depends on a blend of factors: (1) Various depth cues can be used in the model such as shading, shadows, fog and landmarks (or known iconic objects), and can help with comparative distance judgements as well as absolute distance judgements; (2) the type and quality of the renderer used can affect the perception of the world. For instance, users’ perceptions may underestimate and be less accurate on non-photorealistic and sketch-based renderings (Gooch and Willemsen 2002; Roussou and Drettakis 2003); (3) interaction methods and the speed of rendering and the speed of interaction affects the users’ perception of immersion in the world; (4) the display device and hardware interface, such as three-dimensional displays (Watt et al. 2005); and (5) our individual human capabilities or disabilities.

6.1 Perspective, depth sensation and field of view in the Golden Square model

Perspective drawings help the viewer relate to a specific distance away from the drawing surface: objects are scaled relative to the viewer. An object is not by default scaled evenly, a circle can appear as an ellipse and a square as a trapezoid. The distortion is known as ‘foreshortening’.

In *SL* the foreshortening effect is exacerbated by the lack of depth of field cues. Perspective experience is limited by the attributes of the graphical images, for the most part without reflection (in *SL* only water has reflection), missing focus, insufficiency of semitones or dynamic shading. It is currently difficult to create a high-fidelity virtual reality world due mainly to technical limitations on processing power, image resolution and communication bandwidth. In time these limitations are expected to be resolved as processor power, image rendering and data communication improve in tandem with production costs. In the meantime, a compromise has been made between image quality and object functionality. The games industry with its predominantly fixed-scene environment have better graphical rendering than their counterparts in the more interactive virtual worlds such as *SL*, where spontaneous object creativity and user-built environments have greater functional demands and higher priority.

The ‘Draw Distance’ controls how far the user can see. Increasing the ‘Draw Distance’ improves realism but reduces performance. When this distance doubles, the amount of data that must be downloaded and displayed goes up between four and eight times (area or volume). Polygons beyond the draw distance are not drawn. *SL* allows users to manually set the Draw Distance in the range 64–512 m to adjust and balance performance and graphic quality. Graphics quality and realism of the scene increases with draw...
distance, but the overall performance or frames per second (FPS) will decrease. The number of FPS can be affected in SL if rendering demand for avatars (and or objects) and performance settings are both high.

Various world effects can be used in SL to improve the perception of realism. These include atmosphere, water, lighting, clouds and fog. The water, cloud and sky settings can be fine-tuned in the SL client viewer to affect the perception of realism in the environment. Transitions can be set and will follow sunrise and sunset intervals of the day. How clouds appear, atmosphere and lighting can be adjusted to enhance or dramatize the perception and realism of a scene. The settings are local to the user. In the SL client viewer (from the ‘World’ drop-down menu, by using ‘Environment Settings’ or ‘Environment Editor’) in the Advanced Sky window, changes can be made to atmosphere, lighting, clouds and fog. Dramatic changes can be made to the intensity of colour and hues, to affect distance horizon views and simulated altitude change variations.

7. Lessons learned

In developing the Golden Square Project we created a three-dimensional virtual model in SL that mimics the real world. There are many principles and lessons that can be extracted from our experience. We present here several lessons that, on reflection, we believe form good practice for developing a scale model in a virtual world, and in particular in SL. The following can act as a list of guidelines for users wishing to develop scaled environments. These guidelines follow the order of this article and act as a summary of the main ideas.

Production

1. Gather materials. Both holistic information and detailed information should be gathered. If the scaled environment is to have ‘size constancy’ the individual objects need to be placed in appropriate contexts.

2. Estimate the overall size of the build. Will it fit into the environment? Is it feasible to build in the timescale given?

3. Develop appropriately scaled scaffolding that will enable the environment to be easily built. Aerial photographs or maps can be used to create a ground plan; lay out markers on the floor and scaffolding to guide the height (silhouette). Create necessary reference scale tools.

Physical Environment

4. Develop your model with several known objects (landmarks) at a variety of positions in the scene that can be used to make size-constancy judgements, for example, well-known objects such as benches, waste bins and telephone boxes.

5. Include external building features such as recessed windows and doorways, concrete tiles on the floor, protruding pillars and steps leading up to the doors that can be used for absolute judgement perception.

Perception
6. Use the longest ‘draw distance’ that is possible on the renderer and encourage the use of environment effects such as fog or clouds.

7. Change the viewport to ‘mouse look’ view, and possibly restrict the user to walking (and not flying) – if the world is to be used for situation-aware tasks.

8. Conclusions

We believe that scale should be considered at every stage of the production, physical environment design and perception of virtual environments. There are specific challenges and choices for the developer to consider for each of these categories. Each will dictate or frustrate the expectations of the viewer. Although it may be possible to change the size of an avatar in SL, we believe that it is difficult in practice to get users to resize their avatars, and this may act as an inhibitor, discouraging users to visit a site.

Moving through realistically scaled three-dimensional spaces will provide new opportunities beyond the bounds of real-world constraints. In addition, the use of novel interface technologies will further enhance users’ experience. We suspect, however, that most designers are still thinking in classical dimensions. Although three-dimensional environments have only been in popular use for the last eight years, it is certain that the next ten years will see a definite change, merely considering that a generation of gamers who are familiar with three-dimensional worlds – four million 15-year olds – will soon come into the research, design and development work places of the future to take this emerging technology forward.

Acknowledgements

The authors wish to thank and acknowledge John Owen for his advice and help in developing our SL modelling skills, and Nigel John for his support and help during this project.

References


**Contributor details:**

Dr Jonathan C. Roberts is a Senior Lecturer in the School of Computer Science, Bangor University, UK. His research is focussed on Visual Analytics: the analysis, visualization and interaction of data. Particularly, he investigates methods that allow information to be displayed and explored, how different users’ opinions can be discussed and included in visual depictions, and how different versions of the data (such as from changing parameters on a mathematical simulation) can be interacted with. He has researched solutions including multiple coordinated view desktop systems, virtual environments and new technologies such as haptic and force-feedback devices. He currently organizes projects in flood risk resource management, coastal erosion and visualization, information visualization and visual analytics, and VRLink SIG (connecting businesses through virtual reality).

Dr Robert Gittins was the Regional Manager for ITWales at the Bangor University office until December 2008, and a Lecturer in Computer Science, Bangor – recently Generic Support Manager for The Research Institute of Visual Computing, and now the Project Development Manager for the A4B Project VRLink SIG. His research interests are in the area of software development methodologies and virtual environments. He is a member of the British Computer Society, and in particular the BCS Industrial Liaison Officer for the North Wales & Chester Area. He is a founder member of the BCS Animation and Games Development Specialist Group. He founded the ITWales ‘Software Developers Special Interest Group’ (Bangor & Swansea) across business and academia and is
responsible for brokering industrial projects for undergraduate and postgraduate students. He has lectured on the subject of virtual environments and led workshops on the use of virtual environments to minority groups, business groups and people with limited mobility.

Dr Rhys Thomas obtained a B.Sc. in Computer Science at Bangor University in 2005 and subsequently studied for a Ph.D. funded by the Engineering and Physical Sciences Research Council (EPSRC). His main area of research is based around Augmented Reality as a tool for anatomy education. In July 2009, Rhys took up a research fellowship at Leeds University to work on a virtual reality microscope for diagnostic pathology.

**Contact:**
School of Computer Science, Bangor University, Dean Street, Bangor, LL57 1UT, UK.
e-mail: j.c.roberts@bangor.ac.uk.

School of Computer Science, Bangor University, Dean Street, Bangor, LL57 1UT, UK.
e-mail: rgittins@bangor.ac.uk.

School of Computing, University of Leeds, Leeds, LS2 9JT, UK.
e-mail: scsrgt@leeds.ac.uk.