

A 3D Lighting System for fMRI Rendering Fidelity Experiments

Giannis Christodoulou Dept. of Electronic and Computer Engineering Technical University of Crete, Greece jchrist@ced.tuc.gr	Eugenia Radulescu Brighton and Sussex Medical School Sackler centre for Consciousness Science University of Sussex, UK E.Radulescu@bsms.ac.uk N.Medford@bsms.ac.uk H.Critchley@bsms.ac.uk	Nick Medford Hugo Critchley Phil L. Watten Dept. of Informatics University of Sussex, UK p.l.watten@sussex.ac.uk	Katerina Mania Dept. of Electronic and Computer Engineering Technical University of Crete, Greece k.mania@ced.tuc.gr
--	---	--	---

Abstract— Although improvements in basic computer graphics rendering hardware and lighting algorithms have produced some remarkable results, it is still computationally demanding to render a highly realistic Virtual Environment (VE) in real-time. This paper presents a real-time synthetic lighting system incorporating sophisticated global illumination algorithms aiming to induce similar subjective lighting impressions as in the real world. The lighting system proposed is designed to render an interactive VE on an fMRI display, enabling the conduct of formal neuroscientific experiments, investigating the effects of visual fidelity as well as varied lighting configurations on feelings of presence, ‘reality’ and comfort. Ultimately, the goal is to use this system to explore the effect of lighting variations (daylight vs forms of artificial light) on subjective impressions of a group of patients suffering from the ‘depersonalization’ syndrome. The system was developed in close collaboration with the Brighton and Sussex Medical School in the UK. It was a challenge to develop an interactive lighting system to be utilized for fMRI experimentation due to infrastructural and technical demands. Such demands were based on acquiring user input when immersed in the constrained environment of an fMRI scanner while the system reacts to it in real-time. fMRI experiments usually employ simple display material, for example using photographs, video clips or simple computerized stimuli. Employing VEs in fMRI has the advantage that it is possible to involve participants in *interactive* animated environments which more realistically reflect social and emotional situations.

Keywords — Lighting simulation, presence, neurocorrelates of fidelity.

1. BACKGROUND

While there has been extensive work to enhance the perceptual realism of VEs and graphic displays, it is never clear to what extent such improvements in visual quality are associated with differences in visual cognition of the trainees (Mourkoussis et al. 2010). Recent computer graphics technology allows high quality graphics to be produced – but how much could the quality be reduced without consequent changes in the cognitive processes in the human observer? This may depend on the importance of aspects of a scene such as shading, lighting, geometry etc. and its scope which drives human perception at that instant. One of the purposes of the work described here is to understand how the brain processes lighting propagation in real and 3D scenes.

Before we are able to deal with such issues, we will need to understand more about processes of visual cognition when viewing dynamic scenes, therefore, such work is truly interdisciplinary. The well-known work of Land and his colleagues on making a cup of tea, driving, or playing music, suggests that vision is used to guide action on a “just in time” basis, and that very few fixations are made to task-irrelevant locations in the scene (Land et al 1998). However, this work does not allow us to say how such processes would be impaired if the quality of the scene (either its resolution or its temporal properties) was to be altered or somehow impaired or how close a simulation for training is to the real-world task situation simulated (Mourkoussis et al. 2010, Mania et al. 2006). Moreover, we lack knowledge of how different scenes, and observers with different internal states or perspective, would be affected by the relative importance of different aspects of the scene. This knowledge could drive the rendering of a simulation in real-time and ultimately allocating rendering resources on significant aspects of the scene depending on the task situation.

The concept of ‘presence’ has become an issue of much debate during the evolution of VE technology. Presence can be defined as the propensity of a participant to respond to virtually generated sensory data as if they were real (Sanchez-Vives et al. 2005). This definition encompasses not only subjective conscious experience of the VE, e.g. the feeling of ‘being there’, but also unconscious aspects such as autonomic and other physiological responses. There is now extensive literature on presence and VEs,

and it is known that changes in VE features and attributes, e.g. lighting effects, depiction of motion, can exert a strong modulatory effect on different aspects of presence and fidelity of a simulation (Slater et al. 2009). VE technology thus has considerable potential as a tool for exploring experiential aspects of feeling states and the biological correlates of these states.

The obstacles to a proper understanding of human perception of synthetic scenes are many. Visual cognition is a complex process, involving the encoding of low-level features, interpreting the gist of the scene and selecting aspects of the scene to interrogate during an active scanning process. No single discipline can hope to tackle these big questions. We therefore propose a truly interdisciplinary approach of measuring the simulation fidelity and subjective ‘sense of being there’ or presence of a synthetic system by utilizing neuroscientific data through fMRI imaging. Such experiments will indicate the relative ‘fidelity’ of a synthetic scene in relation to whether such cognitive or neuroscientific information resemble the cognitive/neuro patterns of behaviour identified in real-world situations. Such processes may not only establish a cognitive fidelity metric for training simulations but also have implications for the therapy of disorders such as symptoms of depersonalization-derealization, as well as advance cognitive neuroscience research.

People tend to respond realistically to events within synthetic environments and even to virtual humans in spite of their relatively low fidelity compared to reality (Mania et al. 2010). For example, VEs have been used in studies of social anxiety and behavioural problems and individuals with paranoid tendencies have been shown to experience paranoid thoughts in the company of virtual characters (Freeman et al. 2008). These provide specific examples of ‘presence’ – the tendency of participants to respond to virtual events and situations as if they were real. One recent study suggests that people interact with virtual characters in a realistic and emotionally engaged way (Slater 2006). This study reported a scenario in which experimental participants took part in a virtual version of the Milgram experiment, in which people were asked to administer increasingly severe punishments to virtual characters performing a memory task (Milgram 1963). Participants showed autonomic responses which were consistent with states of intense emotional arousal, as would be expected if the experiment had used real participants. Thus it is evident that people are able to emotionally engage with synthetic spaces and virtual characters as if they were real. Recent work explores neural correlates of empathy and believability of synthetic characters by implementing an economic game combined with Milgram’s original experimental scenario (Figure 1) which can be interactively played in an fMRI scanner (Rivera et al. 2010).

Previous neuroscientific work involving synthetic scenes has been limited to situations where specific reactions were recorded rather than compare internal emotional processing in the real-world and in synthetic simulations (Lee et al. 2004). The goal of the planned experiments described in this paper is to assess the fidelity of a simulation as well as develop imaging systems which will ensure that controlled fMRI experiments involving patients are possible.

Previous computer science research into evaluating whether virtual characters in immersive collaborative environments fulfill their role or whether synthetic scenes induce a sense of ‘reality’ is limited to acquiring ratings of pleasantness, presence or user fidelity impressions through self-report after the experience has occurred (Vinayagamoorthy et al. 2005). The ultimate goal of the system presented here is to explore whether natural and artificial scenes of varied fidelity for training or for therapeutic purposes engage common perceptual or neuroscientific mechanisms. Such input is non-obtrusive and is derived at the same time as the experience occurs.

It is not straightforward to provide synthetic stimuli to be utilized for fMRI experimentation due to the infrastructural and technical demands of acquiring user input while immersed in the constrained environment of an fMRI scanner while the system is expected to react to it in real-time. fMRI experiments usually employ simple display material, for example using photographs, video clips or simple computerized stimuli (Lee et al. 2004). Using VEs in fMRI has the advantage that it is possible to involve participants in interactive animated environments which more realistically reflect social and emotional situations. This seamless naturalism and interactivity is impossible to achieve with video clips. In addition, certain experiments could be conducted using synthetic rather than real-world stimuli for ethical reasons. Because of the advantages that VEs offer for fMRI and also the increasing cost-effectiveness of the application of VEs, it seems very likely that in the next few years the use of VEs in cognitive neuroscience will undergo a rapid growth. In the following sections, we will present an interactive lighting system developed and installed in an fMRI scanner implementing a formal neuroscientific protocol.



Fig. 1: Interactive 3D game displayed in the fMRI scanner by Rivera, Watten, Holroyd, Beacher, Mania, Critchley, 2010.

2. SCOPE

The general aim of the work presented was to build an interactive lighting system installed in an fMRI display. The goal was to experimentally explore changes in regional brain activity associated with changes in the sense of ‘presence’ and reality perception of space and self as well as on subjective feelings associated with lighting such as comfort when immersed in simulations of varied lighting configurations and rendering quality. It was intended to discover more about the neural circuitry that supports such feelings by devising behavioural fidelity metrics of simulations based on neural activity. Another goal was to examine the interaction between changes in the sense of presence and the neural response to, and subjective experience of, emotional material. This is important, because it is known that in mental states where people feel disengaged from their environment (i.e. not ‘present’), emotional responsivity tends to be reduced (Medford et al. 2005).

The purpose of the lighting system presented in this paper was to enable neuroscientific experiments examining how certain aspects of the experience of being exposed in a synthetic simulation such as rendering of lighting propagation and extreme variations of rendering are modified by changes in the environment. This is mostly relevant to a psychiatric syndrome called the ‘depersonalization syndrome’ (Medford et al. 2005). Depersonalization is a state in which a person feels that they and their surroundings are oddly unreal and dreamlike. Most people have experienced mild depersonalization at times e.g. when jetlagged or very tired, or when under stress. Usually it passes within a few hours. However, in some people it can become intense and long-lasting, to a point where it becomes distressing and interferes with daily life. In these circumstances it may be considered as an illness (primary depersonalization disorder). It may also occur in association with other psychiatric illnesses such as depression. The work presented in this paper was based on the observation that people suffering from the depersonalization syndrome often describe feeling slightly ‘altered’ under different lighting conditions. In particular, it has been noted that depersonalization occurs more commonly when people are in artificial lighting conditions than in natural light (Sierra 2009).

In healthy people, it is a common experience to feel that changes in lighting and other features of the environment can induce a sense of not being fully present, or of things not being fully real. This is essentially a mild, short-lived experience of depersonalization. Studying what is happening in the brain during these experiences can tell us more about how the brain creates the sense of reality and the sense of being present or conscious in the world. This has relevance not only to understanding states of depersonalization, but also to understanding states where a person’s sense of reality has become radically altered, as happens in psychotic disorders such as schizophrenia (Sass & Parnas 2003).

Experiential reports of VEs aid the understanding of how aspects of an environment influence feelings of both presence and detachment. This is consistent with reports that being under certain types of lighting induce feelings of ‘unreality’ in individuals vulnerable to dissociation. In this paper, a 3D interactive lighting framework is put forward, rendering interactive VEs in an fMRI scanner of varied physics-based daytime lighting as well as artificial configurations of lighting. The system supports a formal neuroscientific experimental protocol enabling experiments investigating the effect of lighting on neural activity related to presence and depersonalization. Following a strict experimental protocol, the lighting system implemented involved interactive photorealistic synthetic scenes of varied lighting configurations and high vs low rendering fidelity, simulating day-time, night-time and artificial lighting presented to healthy volunteers. They could subsequently interactively manipulate such scenes in the scanner and answer questions set by the psychiatrists collaborating in this project in relation to feelings of presence, comfort and sense of reality. The fMRI scanner acquired brain images at strictly specified timings while the participants followed the experimental protocol, performing the tasks assigned to them while being immersed in the VEs such as navigating, or viewing emotional images. Meanwhile, participants’ physiological measures were acquired, such as heart rate and heart pulse oximetry. In this paper, we focus on the technical description of the system.

3. EXPERIMENTAL PROTOCOL

The lighting framework presented was designed specifically to render the VEs at different times of night and day as well as lit by varied artificial lighting such as fluorescent, incandescent etc. being presented on the fMRI display. It was developed to maintain the imposed time limits and was synchronized with the fMRI scanner. It was designed to simulate both high-fidelity virtual scenes and photorealistically lit scenes including wireframe objects (Figure 2). Most importantly, it allowed the manipulation of brightness of artificial light types in real-time in both cases, providing a real-time simulation of the lighting effects. The system allowed the investigation of the participants’ neurocorrelates of fidelity at the same time as being immersed in the synthetic scenes and in real-time.

Study participants immersed in the fMRI scanner freely navigated the VE displayed on the fMRI display, depicting the inside of a house and, connected to it, a backyard/garden (Figure 2 and 3). By using button boxes, they were able to explore the VE, alter the lighting conditions, look at ‘indoors’ (lounge) and ‘outdoors’ (garden) parts of the environment and interact with certain objects in the scene. They were at intervals asked to give ratings of how they were feeling and how ‘real’ the VE appeared to them. At the same time, the scanner recorded data on brain activity in order to determine whether feelings of ‘presence’ or ‘reality’ were modulated by environmental (i.e. illumination) factors of the VE.

During the first stage of the neuroscientific protocol, a random virtual scene either rendered photorealistically or including wireframe objects, indoors or outdoors was displayed and the participant could freely navigate it for a specified time period.

Subsequently, the system randomly selected the virtual scenes representing the same time of day as this first scene, then moved on through the following sequence:

Morning → Midday → Afternoon → Evening → Morning → ...

After viewing each scene, a range of questions was presented to the participant assessing perceived sense of presence, sense of ‘reality’ and comfort.

The sequence of virtual scenes was repeated during the second stage of the experiment, with the exception that no wireframe scenes were displayed. A selection of images was presented to the participants, who rated them on simple visual analogue scales for arousal and valence. The participant watched nine pre-defined emotional images superimposed on a TV set placed in each synthetic scene, three from each category of pleasant, neutral and unpleasant respectively. Although the same emotional images were displayed in each virtual scene across participants, the order that they were displayed was randomized and the resulting sequence of emotional images was produced and saved in the specified *Images* array.

During the 3rd experimental stage, a range of artificially lit, photorealistically-rendered evening indoors scenes were displayed either in full detail or including wireframe objects. The lighting configuration switched randomly to one of the available light types, i.e. 40 watt tungsten, 100 watt tungsten, halogen, carbon arc, standard fluorescent and cool white fluorescent light types. While an artificial lighting configuration was presented, the participant was asked to change the brightness to their most comfortable value by adjusting a scale from low to high brightness, or vice versa. Participants rated each scene in relation to perceived presence and comfort.

4. IMPLEMENTATION

A system is proposed which simulates realistic lighting and allows the participant to interactively manipulate it while immersed in a fMRI scanner. The 3D lighting framework which incorporated the experimental protocol described in Section 3 was developed with the Unreal Development Kit (UDK, <http://www.unrealengine.com/udk/>). UDK consists of different parts, making it act both like a game engine and a 3D authoring environment. It provides the necessary tools to import 3D objects, create and assign materials on objects that affect the lighting distribution, pre-compute lighting effects and import and use sounds and sound effects. It, also, allows the designed application to seemingly attach to Flash User Interfaces (UI). UDK can also be used to render the created VEs, as well as create and respond to events while navigating the synthetic scenes. The main components inside UDK are the Unreal Editor, which is used to create and edit VEs, handling all the actors and their properties located in the VEs, the Unreal Kismet, which allows for the creation of sequences of events and corresponding actions and the Unreal Matinee, responsible for the animation of actors or real-time changes in the actors’ properties. The UDK platform enabled the development of interactively manipulated photorealistically-rendered synthetic scenes, as well as providing the necessary tools to overcome the technical difficulties of using interactive visuals in the constrained environment of an fMRI scanner.

UDK offers the ability to use both C/C++ and UnrealScript, which provides the developers with a built-in object-oriented programming language that maps the needs of game programming and allows easy manipulation of the ‘actors’ in a synthetic scene. UnrealScript is purely object-oriented and comprises of a well-defined object model with support for high level object-oriented concepts such as serialization and polymorphism. This design differs from the monolithic one that classic games adopted having their major functionality hard-coded and being non-expandable at the object level. Before working with UnrealScript, understanding the object’s hierarchy within Unreal is crucial in relation to the programming part. The main gain from this design type is that object types can be added to Unreal at runtime. This form of extensibility is extremely powerful, as it encourages the Unreal community to create Unreal enhancements that all interoperate. The five main classes one should start with are *Object*, *Actor*, *Pawn*, *Controller* and *Info*.

Object is the parent class of all objects in Unreal. All of the functions in the *Object* class are accessible everywhere, because everything derives from *Object*. *Object* is an abstract base class, in that it doesn’t do anything useful. All functionality is provided by subclasses, such as *Texture* (a texture map), *TextBuffer* (a chunk of text), and *Class* (which describes the class of other objects). *Actor* (*extends Object*) is the parent class of all standalone game objects in Unreal. The *Actor* class contains the functionality needed for an actor to be placed inside a scene, move around, interact with other actors, affect the environment and complete other useful actions. *Pawn* (*extends Actor*) is the parent class of all creatures and players in Unreal which are capable of high-level AI and player controls when a game is being implemented. *Controller* (*extends Actor*) is the class that defines the logic of the pawn. If *Pawn* resembles the body, *Controller* is the brain commanding the body. Timers and executable functions can be called from this type of class. *Info* (*extends Actor*) is the class that sets the rules of interaction or gameplay. Users joining will be handled in this class, which decides which *Pawn* will be created for each user and which *Controller* will handle the behavior of the pawn.

Unreal Lightmass utilized for rendering is an advanced global illumination solver. It uses a refined version of the radiosity algorithm, storing the information in each illuminated 3D object's light map, while providing ray-tracing capabilities. Support for ray-tracing is based on Billboard reflections, which allows complex reflections even with static and dynamic shadows with minimal CPU overhead. Unreal Lightmass is provided as part of UDK and it can only work on scenes created through it. Its performance is dependent on the complexity of the scenes created and the types of light emitting sources that exist in the scene. It is optimized to increase the renderer's performance.

Creating the visual content

A five by five meters room connected through a door to an equally sized yard was selected as the rendered displayed environment, allowing for the simulated light effects of a sun and of an artificial light inside the room to be seen when the viewpoint is set indoors and outdoors respectively. The indoor room and the outdoor yard were designed to be as similar as possible, in terms of the 3D objects placed in them. Wireframe versions of the photorealistic scenes were created by using wireframe-specific materials. The created 3D objects were imported into UDK, by selecting the 'import' option in the asset library of the Unreal Editor. UDK reads the file containing the exported 3D objects and recreates the geometry of the objects. It also creates the different material slots for each part of the 3D object and initializes the UV channels of the object.

Collision detection is automatically performed by UDK, after assigning a collision vector to each 3D object according to its geometry. For the purposes of the system described here, the collision detection mechanism was required to simply prevent the participant from intersecting other objects, or passing through walls.

Lighting effects greatly enhance the photorealism of the synthetic scenes. UDK's Lightmass was used to pre-compute photorealistic lighting effects for outdoors (Figures 2,4,6) and indoors (Figures 3, 5, 7) during the morning, mid-day and afternoon as well as indoor artificial lighting ranging from 40W/100W Tungsten, Halogen, Carbon Arc, Standard Fluorescent and Cool White Fluorescent light types (Figures 9-14). The configuration settings are listed in Figure 8.



Fig. 2: Screenshot of the Morning Outdoors Scene (left) and the respective wireframe version (right).



Fig. 3: Screenshot of the Morning Indoors Scene (left) and the respective wireframe version (right).



Fig. 4: Screenshot of the Midday Outdoors Scene (left) and the respective wireframe version (right).



Fig. 5: Screenshots of the Midday Indoors Scene (left) and the respective wireframe version (right)



Fig. 6: Screenshots from the Afternoon Outdoors Scene (left) and the respective wireframe version (right)



Fig. 7: Screenshots from the Afternoon Indoors Scene (left) and the respective wireframe version (right).

Light Source	R G B Values	Color
40W Tungsten	255, 197, 143	Orange
100W Tungsten	255, 214, 170	Light Orange
Halogen	255, 241, 224	Light Yellow
Carbon Arc	255, 250, 244	Yellow
Standard Fluorescent	244, 255, 250	Cyan
Cool White Fluorescent	212, 235, 255	Blue

Fig. 8: Artificial light types and their respective RGB color value.



Fig. 9: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *Standard Fluorescent* artificial light type applied.



Fig. 10: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *40 Watt Incandescent Tungsten* artificial light type applied.



Fig. 11: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *100 Watt Incandescent Tungsten* artificial light type applied.

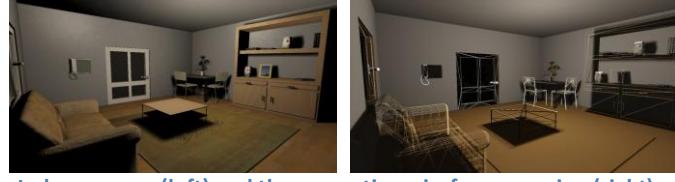


Fig. 12: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *Halogen* artificial light type applied.



Fig. 13: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *Carbon Arc* artificial light type applied.



Fig. 14: Screenshots of the Evening Indoors scene (left) and the respective wireframe version (right) with the *Cool White Fluorescent* artificial light type applied.

UDK provides the option to create a *Light Actor* including a custom set *Light Color* as an RGB value, as well as the ability to change this value in real-time. In order to make the indoors light actor to simulate the required artificial light types, their RGB value was approximated according to Hastings (2011) and 3ds Max 2011's representation of artificial light types. The RGB value associated with the Standard Fluorescent light type, which was the initial light value, was used to define the *Light Color* option of the *PointLightToggleable* Actor properties.

The image seen by the participant from within the fMRI scanner was rear projected through a collimating lens on to a rear-projection screen. The collimating lens can add artefacts to images with very high spatial resolution, but this was not a problem in this case. However, the rear projection screen did add a yellow cast to the entire image. This was corrected using the low-level manual colour correction built into the graphics card so that the colour of the projected image matches a calibrated monitor. Digital gain in the projection system can cause the white levels to clamp at a value less than 1.0. This effect was removed by disabling image processing in the projection system and limiting the overall white-level at the output stage of the graphics card.

Handling the Stages of the Experiment using States

The experiment was performed in three stages as described in Section 3. A *Scene Controller* kept record of the active stage of the experiment linked to each loaded synthetic scene and generated the respective events that were required for each specific experimental stage.

The *Controller*'s experiment flow control was designed to be managed through different states, each one attached to a specific stage of the experiment. The following variables were declared to be saved to the *Controller*'s configuration file and were loaded each time the *Controller* was instantiated:

```
var config array<int> Scenes; // The sequential order of displayed scenes
var config int CurrScene; // The scene that was currently being displayed
var config int CurrStage; //The active stage of the experiment (between 1, 2, 3);
var config array<int> Images; // The emotional images displayed, in a specific order
```

```

var config int CurrlImage; //The specific image displayed
var config int ParticipantNumber; //The auto-incremental participant number used to differentiate log file names.

```

When a new experiment started, the *StartExperiment* method in the *Controller* class was executed. This method initialized all the experiment variables and then loaded the first 3D scene.

The *initializeExperiment* method initialized the *Scenes* array, which contained the synthetic scenes to be loaded in their specific order. It also initialized the *Images* array, which contained the emotional images to be displayed during the second stage of the experiment. Whenever a new 3D scene was loaded, a new Controller was instantiated and loaded its variables from the configuration file. Then, it entered the *FindCurrentExperimentStage* state, which checked the variables signifying the experiment's stage and the 3D scene that was currently rendered and set its state accordingly. From there on, the *Controller* could initialize the timers and the events needed for the specific stage of the experiment and plan the flow of events for that stage accordingly. Each virtual scene could listen to all types of events, referring to all stages of the experiment and react to them, however the Controller would only generate events for the current stage of the experiment.

Interaction and User Interfaces

The core of the development of the complete application was implemented in UnrealScript. Several classes were created in UnrealScript which handle the aspects of the application's rendering, navigation and interaction with the synthetic scenes.

One of the main requirements of the application was that the synthetic scenes were to be interactively navigated and that the application was required to react to user input, ranging from navigation of the synthetic scenes to interaction with the questionnaires. In order to achieve this, the buttons corresponding to the physical button boxes placed in the fMRI scanner and utilized for interacting with the scene were registered, with their respective commands. These are shown in Fig. 15.

One major issue arising from the button boxes hardware is the fact that they did not report how long each button was pressed rather only that it was pressed, no matter if it was being held pressed for a specific amount of time. This was due to the fact that the hardware drivers were reporting the “button released” message immediately after the “button pressed” message to the operating system. Because of this attribute of the button boxes, each participant immersed in the fMRI scanner could not keep a button pressed to, for instance, navigate the scene, e.g. keep the “Move” button pressed to keep moving forward. In order to address this issue, only the “button released” messages were being monitored. The participant was instructed to press and release a button to start an action and then press it again to stop it. So, for example, if the participant wanted to move forward in the virtual scene, the “Move” button was pressed. Subsequently, the viewpoint moved forward until the user repressed the “Move” button. The navigation in the virtual scenes was simulated between the “start” and “end” instructions of the participant. For this implementation to go through, there were five boolean variables declared in *VRExperimentPlayerController*:

```

//variables used to handle participant's navigation
var bool turningLeft;
var bool turningRight;
var bool lookingUp;
var bool lookingDown;
var bool movingForward;

```

Each method registered in the input configuration file handled a type of movement that was designed to reverse the value of its corresponding boolean variable, while disabling the other possible moves. The main method that handled the Pawn’s movement when freely navigating the synthetic scene was *ProcessMove*, which was called periodically and automatically by UDK. This method checked the Boolean variables and if they were true, the Pawn performed the respective movement with its predefined speed. For example, in order to check if the Pawn should move forward, the *movingForward* variable was checked and then the Pawn’s *Acceleration* was adjusted according to its rotation inside the scene as well as the predefined speed in the Pawn’s class, in its *default properties* block. A start-up screen including instructions was presented to participants when inside the scanner, as shown in Figure 16.



Fig. 15: Photo of the Current Designs HHSC-2x4-C response pad used in the experiments.

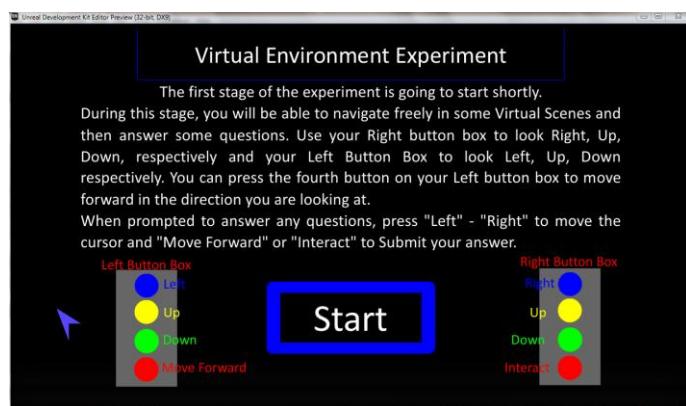


Fig. 16: The start Flash menu displayed when the application was initiated.

Logging of events and participants' actions

The application recorded every event as well as every action by the participant in a separate log file dedicated to each experimental stage. Each generated event changing the current state of the experimental protocol such as the loading of a new scene, or moving from free navigation in a scene to answering the questionnaires was recorded in the log file. Each log entry reported the specific event that occurred and the exact time it happened. The time was measured in milliseconds; the beginning of the current stage of the experiment was assumed to be time point 0.

In order to implement the log file operations, a .dll file was bound to the *Controller* class, providing the necessary methods to record each log entry. This was implemented because UDK's support for I/O operations is limited avoiding extreme overhead for the application since UnrealScript is very slow and inefficient for such operations. Whenever a new log entry was recorded in the log file, the controller could simply call the C/C++ function residing in the .dll file and let it perform the operation.

Every action that the participant performed while being presented with a virtual scene was also recorded in the log file, as well as the participants' responses to the questionnaires. The viewpoint of the participant was recorded by inserting a log entry every time the object in the center of the screen changed. Initially, the tracking of the participant's movement in the scene was performed via the *ProcessMove* function and a log entry was created whenever the *movingForward* variable was true triggering the Pawn to move forwards polling the new location of the pawn in the 3D scene. However, this proved to be extremely inefficient since the *ProcessMove* function was executed several times per second. This approach led to a vast amount of log entries referring to participant movement in the scene, even for a very small and limited move. Also, it proved to be difficult to translate later the exact location of the Pawn used from the participant from the recorded XYZ coordinates. This issue was addressed by dividing the room and the yard into nine equally-sized numbered square zones and assigning a Trigger actor in the center of each zone. The participant's movement from one zone to another was continuously recorded.

Manipulation of the indoors artificial light

During the third stage of the experiment, it was required that the artificial light residing in the evening indoors scene would change dynamically and in real-time in terms of color and brightness, based on user interaction. This feature was challenging because real-time rendering of lighting propagation was required. UDK's Lightmass pre-computed the default lighting effects and the dynamic changes affected these pre-computed effects. The artificial light corresponded to a Standard Fluorescent light by default. During the third stage, participants were shown a random lighting condition for fifteen seconds and subsequently answered some questions inquiring about their impressions of the lighting and then moved to the next lighting condition. When the timer expired and a new lighting condition was presented, an event signaling the specific light type to be displayed next was activated. The synthetic scene captured that event in Unreal Kismet and caused an action responding to that event, which handled the change of the light's properties, i.e. its color and brightness. The available light types that were randomly presented to the participant were 40 Watt Tungsten, 100 Watt Tungsten, Halogen, Carbon Arc, Standard Fluorescent and Cool White Fluorescent.

When the Controller had to change the artificial indoors light to a specific type, it activated the respective index of the *Change Light Color* event. Then, in the Unreal Kismet in the synthetic scene, that index of the event was connected to a *Matinee*, (responsible for the animation of actors or real-time changes in the actors' properties) changing the artificial indoors light to the corresponding color.

Time Synchronization

The experimental stages were controlled by previously specified time limits. The application was synchronized with the brain images acquired by the scanner. Time limits imposed were the time available to the participant to navigate a scene before they had to be presented with a questionnaire, or the time the participant had available in order to answer a questionnaire before the next presented scene. The *Controller* class was developed in order to control the time limits and react so as not to exceed them. Two of its properties saved and loaded from the configuration file were an index to the current stage of the experiment and an index to the scene currently loaded and rendered. Upon initialization, when a new scene was loaded and visible, the *Controller* checked these properties to identify the current stage of the experiment and adjust its timers accordingly. The *Controller* class used Timers to control the time limits. It created a new timer to count up to the amount of time before an event had to occur. When a timer expired, it called the specific method that simulated the event required to happen. For example, in order to restrict participants' navigation time before they were presented with a questionnaire, the *Controller* class created a timer.

The experiments were conducted inside an fMRI scanner and the application was required to be perfectly synchronized with the scanner, in order to be able at a later stage to identify the exact state of the application associated with each brain image acquired by the scanner. The required synchronization between the fMRI scanner and the application was achieved by sending such sound sync pulses to be recorded as an analog spike signal, whenever an important event occurred in the application while at the same time recording that event and the exact time it happened in the log file. Moreover, the heart pulses and the heart pulse oximetry of the participant were also recorded. The sound sync pulses were directed through the left audio channel of the application, leaving the right audio channel available to the participant. The log file described the state of the application for a specific brain image by the sound sync pulse that was sent last before receiving that image.

5. USER INTERFACES

Although UDK includes preliminary support for User Interfaces (UI), embedding Flash User Interfaces was appropriate in relation to the requirements of the experiments, since the UIs were required to be interactive and to be displayed transparently on top of the rendered scene. These requirements could only be fulfilled with the use of embedded Flash UI applications. The UIs included besides navigation of the 3D scenes, the initial menus displaying instructions before the start of a new stage of the experiments; the embedded questionnaires inquiring the participants' impressions of the virtual scene and the emotional images; the UI which enabled participants to alter the brightness of the artificial lighting configurations during the third stage of the experiment. Each Flash UI was designed so as to be displayed on top of the currently rendered virtual scene. The participant's ability to navigate and look around the virtual scene was disabled every time a Flash UI was displayed, so that the input could be captured by the Flash UI itself.

Fig. 17 depicts a Flash questionnaire displayed on top of the virtual scene. When the participant presses the button to submit a response on a scale from 0 to 100, where 0 was marked as "Not at all", while 100 was marked as "Intensely" by moving a cursor, the Flash UI reported the response to the Controller class for it to be recorded in the log file.



Fig. 17: A Flash UI questionnaire being displayed on top of the currently rendered virtual scene.

Several screens displayed useful information concerning the next part of the experiment guiding participants to what was requested to do next. The screens contain graphical representations of the button boxes, explaining the function of each button and allowing the participant to test it and see it activated after each press as shown in Figure 16. The color of the buttons in the graphical representation was the same as on the original button boxes. Next to each of the buttons of the button boxes a short explanation of its function during the next experimental stage was presented. The buttons on the right button box were assigned to turn right, look up, look down and start an interaction with a 3D object, while the buttons on the left button box were assigned to turn left, look up, look down and start / stop moving forward.

In order for UDK to embed a Flash application as a UI and display it, it was required to import the .SWF file of the Flash UI. This was loaded and displayed in an empty scene by connecting the Open GFx Movie action to the Level Loaded and Visible event, which was automatically generated and activated by UDK when the virtual scene became visible. The Open GFx Movie action was provided by UDK incorporating an imported Flash UI as an argument loaded and displayed on the centre of a screen or on a specified surface.

Two questionnaires were administered to the participants at specific instants during the experiments. The questionnaires are displayed on the screen, allowing the participant to still see the virtual scene while answering the questionnaires. The participant is not allowed to navigate the scene while answering the questionnaires, but rather stay still at a predefined spot. The answers are recorded in the log files. The first of the two questionnaires were displayed during the first and the third experimental stages and explored the effect of lighting on participants' sense of reality of objects and self (related to the depersonalization syndrome) as well as participants' sense of comfort and 'sense of being in the displayed scene. The participant was required to indicate the response by moving a cursor on a scale from 0 to 100. The second questionnaire was displayed during the second experimental stage and it contained a single question in order to rate each displayed emotional image on a scale from 0 to 100, where 0 was marked as "Intensely Unpleasant", while 100 was marked as "Intensely Pleasant" by moving a cursor.

The fact that each participant answered each question at a different pace had to be addressed. If the Flash application allowed each participant to move on to the next question when the answer was submitted, it would soon lead to problems with time synchronization. The approach used was that the participants had a predefined time period available to respond to each question. If they submitted their answer before the time available expired, they would see a "Please wait" message. If the available time passed without the participant pressing the submit button, then the participant was assumed to have failed to submit an answer to that question. The Flash UI would still report the latest value that the cursor was placed upon and then move on to the next question. However, it would give the reported answer a negative sign to denote that the participant did not press the submit button.

During the third stage of the experiment, participants were manipulated the brightness of the indoor artificial light. A scale on top of the screen was displayed on a range between 0.1 and 0.9, where 0 was marked as "Low Brightness" and 0.9 was marked as "High Brightness". Whenever a participant adjusted the light brightness by moving the cursor on the scale, the Flash UI reported the change to the Controller class which in turn generated an event signifying that the brightness changed. This event was captured in Unreal Kismet and an action was connected to it, which handled the change of the light's brightness.

6. CONCLUSIONS

This paper put forward the development of an interactive 3D application system to be projected in an fMRI scanner. The application was developed in order to be used for fMRI neuroscientific experiments, exploring the effect of modification of light and emotional stimuli on the feeling of presence and well-being of the normal and ultimately patient populations when exposed to photorealistically-rendered scenes including high visual quality or wireframe objects.

It was a challenge to develop an interactive lighting system to be displayed in fMRI displays due to the infrastructural and technical demands of acquiring user input and provoking real-time feedback to it while immersed in the constrained environment of the fMRI scanner. fMRI experiments usually employ simple display material, for example using photographs, video clips or simple computerized stimuli. Using VEs in fMRI has the advantage that it is possible to involve participants in interactive animated environments which more realistically reflect social and emotional situations.

The 3D lighting framework was developed in UDK, which enabled the development of interactively manipulated photorealistically-rendered synthetic scenes as well as providing the necessary tools to overcome the technical difficulties of using the system in an fMRI display. The implementation of the 3D lighting framework was adjusted to address the challenges arising from the strict experimental protocol, including the creation of the synthetic scenes and the lighting configuration of the light types that were used in the experiments, e.g. the simulation of a sun at different times, as well as the simulation of artificial light types. In order to make the VEs interactive, ranging from navigation to interaction with the questionnaires, the 3D lighting framework responded to the fMRI-compliant button boxes, surpassing the challenge of handling user input.

REFERENCES

- Freeman, D., Pugh, K., Antley, A., Slater, M., Bebbington, P., Gittins, M., Dunn, G., Kuipers, E., Fowler, D., Garety, P. (2008) 'Virtual reality study of paranoid thinking in the general population'. *Br J Psychiatry* 192: 4. 258-263 Apr
- Hastings, J.T. 2011. 'Reproducing Real World Light', <http://planetpixleemporium.com/tutorialpages/light.html>
- Land, MF, Mennie, N., Rusted, J. (1998), 'Eye movements and the roles of vision in activities of daily living: making a cup of tea', *Inv Ophth Vis Sci* 39 p. S457.
- Lee, S., Kim, G., Lee, J. (2004), 'Observing Effects of Attention on Presence with fMRI'. Proceedings of the ACM Symposium on Virtual reality software and technology, 1C, p 73-80.
- Mania, K., Wooldridge, D., Coxon, M., Robinson, A. (2006), 'The effect of visual and interaction fidelity on spatial cognition in immersive virtual environments'. *IEEE Transactions on Visualization and Computer Graphics journal* 12(3): 396-404.
- Mania, K., Badariah, S., Coxon, M. (2010), 'Cognitive transfer of training from immersive virtual environments to reality. *ACM Transactions on Applied Perception*', 7(2), 9:1-9:14, ACM Press.
- Medford N., Sierra, M., Baker, D., David, AS. (2005), 'Understanding and treating depersonalization disorder'. *Advances in Psychiatric Treatment*, 11, 92-100.
- Milgram, S. (1963), 'Behavioral study of obedience.' *Journal of Abnormal and Social Psychology*. 1963; 67: 371-378.
- Mourkoussis, N., Rivera, F., Troscianko, T., Hawkes, R., Watten, P., Mania, K. (2010), 'Quantifying fidelity for virtual environment simulations employing memory schema assumptions'. *ACM Transactions on Applied Perception*, 8(1), 2:2-2:21, ACM Press.
- Unreal Development Kit, <http://www.unrealengine.com/udk/>
- Rivera, F., Watten, P., Holroyd, P., Beacher, F., Mania, K., Critchley, H. (2010), 'Real-time Compositing Framework for Stereo fMRI Displays'. Poster, ACM Siggraph 2010, Los Angeles, USA.
- Sanchez-Vives, MV, Slater M. (2005), 'From presence to consciousness through virtual reality'. *Nat Rev Neurosci*, 6, 332-339.
- Sass L, Parnas, J. (2003), 'Schizophrenia, consciousness, and the self'. *Schiz Bull*, 2003, 29, 427-444.
- Sierra, M. (2009). 'Depersonalization: A new look at a neglected syndrome'. Cambridge, UK: Cambridge University Press.
- Slater, M. (2006), 'A Virtual Reprise of the Stanley Milgram Obedience Experiments', 2006, *PLoS ONE*. 20, 1:e39.
- Slater, M., Lotto, B., Arnold, MM, Sanchez-Vives, MV. (2009), 'How we experience immersive virtual environments: the concept of presence and its measurement'. *Anuario de Psicología*, 40, 193-210.

Vinayagamoorthy, V., Steed, A., Slater, M. (2005), 'Building Characters: Lessons Drawn from Virtual Environments, Toward Social Mechanisms of Android Science: A CogSci 2005 Workshop', pp. 119-126.