www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

DEVELOPING AN EXPERIENCED-BASED DESIGN REVIEW APPLICATION FOR HEALTHCARE FACILITIES USING A 3D GAME ENGINE

SUBMITTED: June 2010
REVISED: September 2010
PUBLISHED: January 2011
EDITOR: Turk Z.

Sonali Kumar, Graduate Research Assistant,
The Pennsylvania State University, University Park, PA, USA
sonali@psu.edu

Matthew Hedrick, Graduate Student
The Pennsylvania State University, University Park, PA, USA
msh5020@psu.edu

Christopher Wiacek, Graduate Student
The Pennsylvania State University, University Park, PA, USA
cjw5027@psu.edu

John I. Messner, PhD
The Pennsylvania State University, University Park, PA, USA
jmessner@engr.psu.edu

SUMMARY: *Virtual Prototypes are increasingly being used during design reviews of specialized buildings such as healthcare facilities. However, most of these virtual prototyping approaches do not allow the reviewers and end users to interact directly, in real time with elements and objects within the virtual model. This paper focuses on a method to combine the use of 3D game engines with the emerging experience based design approach for healthcare facilities to develop a systematic approach to scenario-based design review of healthcare facilities in an interactive virtual environment.*

First, a virtual facility prototyping framework for rapid creation of a scenario based design review system is defined. Next, strategies to implement this framework to develop an Experience based Virtual Prototyping Simulator (EVPS) application are described. Design information workflows were developed and tested between various BIM authoring tools and the Unity game engine that is used for developing the interactive virtual prototype system. Finally, some lessons learned and issues are highlighted that help direct future research and implementation.

KEYWORDS: *virtual prototypes, game engine, healthcare*

REFERENCE: *Sonali Kumar (2011), Developing an experienced-based design review application for healthcare facilities using a 3d game engine , Journal of Information Technology in Construction (ITcon), Vol. 16, pg. 85-104, http://www.itcon.org/2011/6*

1. INTRODUCTION

Research in healthcare facility design strongly indicates that the physical environment greatly impacts end users in issues of safety and overall health quality. Since healthcare facilities are extremely specialized, they involve a diverse range of stakeholders during the design process. This has led to emerging trends and design approaches such as experience-based design and evidence-based design that encourage participation and collaboration with the end users of healthcare facilities. The end users of these facilities such as the healthcare staff, doctors and nurses as well as patients usually have domain specific knowledge that can better inform the decision-making process during design.

Virtual prototypes are being increasingly used during design review since they can be especially useful for the design of specialized building types such as healthcare facilities. Virtual prototyping provides opportunities for a team of project stakeholders to truly experience design alternatives and concepts in the early stages of the design process. While helping in design review, virtual prototyping also allows multiple participants to navigate through the model space and evaluate the design based on various criteria through numerous vantage points within the model. Prior studies have shown the value of using virtual prototypes during design review of facilities such as courtrooms (Majumdar et al., 2006; Maldovan & Messner, 2006), operating rooms, patient rooms (Dunston et al., 2007), as well as indicated the benefits of using them over physical mock-ups (Leicht & Messner, 2009). Moreover, these studies have highlighted the opportunity to enhance end user engagement in instances where team members leverage and communicate their tacit knowledge, enabling collaborative interdisciplinary participation over the early stages of the design development process.

While virtual prototypes are useful for design visualization, allowing the review participants to walkthrough the design of a facility to examine the space, textures and lighting, they are typically implemented in a static manner. Most virtual prototypes do not allow the user to interact directly with the elements and objects within the virtual model. Moreover, visualization in virtual prototypes is a challenge due to the lack of human characters (or avatars) and animations that depict how the facility is used or the tasks performed within the facility. At present, the prototyping process typically lacks a systematic structure or method to allow for task-based scenarios to take place for the review participant. Although a Building Information Model (BIM) of the facility contains geometric and attribute data that is transferred in the virtual prototype, there is no information on the behavior of the components within the facility. For instance, doors modeled in a facility do not swing open in a virtual prototype, despite containing intelligent attribute information such as location of its hinges. As a result, while reviewing, participants either walk through doors or they are not shown in the prototype. This is because animation is still cumbersome to incorporate while developing the prototypes, which can lead to an unrealistic representation of the facilities. Due to these challenges, most current virtual prototypes do not enable people to truly experience the design. The use of game engines can enable participants to interactively perform certain task-based scenarios within the virtual prototype.

This paper presents a method for a more systematic, experience-based design review by project stakeholders. The foremost step was to ascertain the role of an experience-based design approach in the healthcare design industry and investigate how game engines can be leveraged to create an improved virtual prototyping method for interactive design review of healthcare facilities. This paper discusses a framework for incorporating task-based scenarios that take place in a healthcare setting within an interactive virtual environment. Following this, a strategy is put forth to implement the scenario based framework to develop an experience based virtual prototype application along with certain issues encountered and lessons learned.

2. BACKGROUND

Healthcare facilities not only require specialized and complex functions to be performed in them, but they must also address the needs of the end users such as the patients, staff and healthcare practitioners using the facility. This makes them exceedingly difficult to design, build and operate due to the incorporation of the interdisciplinary knowledge and input of various stakeholders such as the design professionals, engineers, facility managers as well as the end users (patients, staff and medical professionals) of the building. Hence, there is a need to create innovative tools and procedures that facilitate high levels of participatory design and allow better visualization of these complex facilities to aid in the decision-making process during the planning and design of these specialized facilities.

2.1 Experience Based Design

Participatory design approaches that involve the end users in the design process are gaining traction in the healthcare facility design industry. There is widely published research that addresses the subject from different perspectives. Some refer to that perspective as experience-based design (Bate & Robert, 2007), evidence-based design (Hamilton & Watkins, 2009; Stankos, 2007), participatory design (Nutter, 1995; Luck, 2003), user-centered design (Norman, 1988) and other terms that appear frequently. Since, users evaluate the built environment differently from designers (Zimmerman & Martin, 2001), participatory design approaches attempt to bridge a gap in understanding between users and designers. This can be extremely useful in understanding the tasks performed by health care practitioners by incorporating their knowledge of how typical daily operations are performed in the healthcare setting into the virtual environment.

Experience-based Design (EBD) is an innovative approach to design that focuses on end user and staff experiences in a facility to identify creative design solutions. While the construction industry is focusing more and more of its attention on hospital and health care construction (Manning & Messner, 2008), the consumer perspective has been influencing critical decisions made by hospital administrators for health environment design. Many studies suggest strong links between the physical environment to the patient and staff outcomes in the following areas: Reducing staff stress and fatigue and increasing effectiveness in delivering care; improving patient safety; reducing stress and improving outcomes; and finally improving overall healthcare quality (Ulrich et al., 2004).

The NHS Institute for Innovation and Improvement uses the term Experience-based design for a user-focused design process with the goal of making user experience accessible to the designers and to allow them to conceive of designing experiences rather than designing services (Bate & Robert, 2006). Similarly, the Center for Health Design is focusing on employing evidence-based design, which they define as “the deliberate attempt to base building decisions on the best available research evidence with the goal of improving outcomes and of continuing to monitor the success or failure for subsequent decision making.” Implementation of evidence-based design requires that design professionals, healthcare planners, and healthcare organization management teams be familiar with the process to follow to identify research, to create hypotheses, and to gather, implement and report the data associated with their projects. Virtual prototypes can help stakeholders envision and better visualize the healthcare facilities and use it as an effective tool for experiencing activities that take place in these facilities.

2.2 Game Engines for Design Visualization

Game engines are the core software component that provide the underlying technology, simplify development and incorporate all the elements vital in a game like physics, collision detection, graphical user interface (GUI), artificial intelligence, network functionality, sound and event engine (Eberly, 2007; Fritsch & Kada, 2004). Most game engines have a built-in physics engine that supports basic physics, collision detection, rigid body and vehicle physics. However, at present there is only an insignificant relationship between game engines and standard architectural or design visualization tools as they seldom offer real-time rendering and simulations that game engines do.

Interactive features are important for users of the virtual environment as it helps them relate what they are seeing in the virtual environment to the real world (Shiratuddin, 2008). For instance, end users can also be placed in a simulation as avatars and then simulate movement-based scenarios and virtually walk through a design. Unlike prescripted walkthrough or flythrough videos, movement is not forcefully prescribed or scripted. This allows end users to interact with the virtual facility and perform collaborative exploration of the designed environment.

Furthermore, game engines allow multiple simultaneous users to explore the designed environment. According to Shiratuddin (2008), with the advances in networking system for online games, one of the strengths of the game engine is its capability for multi-participant networking. The inherent multi-user nature of the game technology lets clients connect to its server using the game's client software over the Internet.

2.3 Simulation of Experience: Scenarios

To truly experience the tasks that are performed in healthcare facilities, the tasks should be simulated as scenarios within virtual prototypes. In the gaming world, the core of Massively Multiplayer Online Games (MMOG) revolves around completing quests or a series of clearly outlined tasks that are given to the player to complete for in-game rewards. Similarly, in the context of experience-based design simulations, specific healthcare tasks can be categorized as scenarios that are movement, task, or inquiry based. The scenarios could vary depending on the user, the type of task being performed and the issues it addresses. For instance a movement-based scenario could involve a nurse moving the patient from the Emergency Department (ED) to the patient room. Large–scale healthcare facilities could benefit through simulation of such scenarios as it could help address issues of way finding in large spaces and also check if there are adequate architectural clearances to move equipment, wheelchairs and patients beds through all the corridors of the facility. Similarly scenarios for design professionals of the healthcare facilities could pertain to spatially reorganizing the architectural model in the virtual environment and evaluating different design options.

To create a framework of user scenarios, it is important to categorize and define the various tasks that can be simulated. It is then necessary to describe the scenarios and identify examples of how these scenarios are carried out and what type of healthcare facilities can benefit the most from them. Furthermore, it is also crucial to identify the participants that can be involved in each type of scenario and the objects required for the simulations.

3. RESEARCH GOALS AND OBJECTIVES

In previous studies that we performed with static virtual 3D stereoscopic virtual prototypes in an immersive display (Maldovan & Messner, 2006) we noticed that the reviewers, and in particular end user reviewers, of the models would frequently be discussing the tasks that they would perform in a space. Yet, we would only be able to navigate around the model and allow them to envision in their mind the task performance. Through their own visualization of the task, they could provide valuable feedback to the design team. Therefore, the goal of for this research was to develop a more interactive virtual environment which allows the user to explicitly perform typical daily tasks just as they would in a physical setting. This should allow the user to improve their design review feedback and expose flaws in the architectural layout. This would also allow the design professionals to enter the design in the role of end users through prescripted scenarios. Moreover, it could help demonstrate incompatibilities between equipment (e.g., wheelchairs and carts) and the architectural geometry, giving the project team the opportunity to revise the design of the building before entering later stages in construction where it is very cost prohibitive to do so. Ideally, the reviewers would be able to navigate a model that is imported directly from various BIM authoring tools. This would ensure a level of architectural authenticity, reduce the time to develop the prototypes making it more feasible to perform multiple reviews, and hopefully eliminate the “game” feel that is frequently prevalent in other virtual mock-ups.

3.1 Objectives

The primary research objective is to define and develop a virtual prototyping framework for the rapid creation of scenario-based design reviews of facilities. This was operationalized by developing a virtual prototyping application that allows interactive design review for multiple stakeholders within the healthcare facility design context. The following objectives have been pursued to achieve the primary objective:

- Develop a framework and approach to structure the healthcare activities into scenarios that can be simulated in an interactive virtual prototyping application.
- Design and develop an interactive computing platform titled the Experience-based Virtual Prototyping Simulator (EVPS) based on the requirements of the healthcare facility.
- Develop the EVPS application for future testing and assessment in both the academic and healthcare setting for conducting collaborative and interactive design reviews of healthcare facilities.

3.2 Research Approach and Methodology

This paper focuses on developing a framework of task-based based scenarios and expanding upon current virtual prototyping methods. The approach is to develop an innovative computing solution within a game engine environment which will allow team members to load and engage in interactive task-based scenarios as represented through the perspective of various users during the early stages of the design process.

The following research steps have been performed to meet the objectives of this research:

1. Develop an information framework for defining interactive scenarios in virtual environments, which includes a task-oriented structure.
2. Select an appropriate programming environment / rendering engine for the development of an interactive virtual environment termed the Experience based Virtual Prototyping Simulator (EVPS).
3. Develop and assess workflows for importing facility models from various BIM authoring tools (e.g., Autodesk Revit) to the interactive virtual environment.
4. Develop an interactive interface to allow the end user to carry out specific task scenarios in the virtual environment through scenario definition, scripting and additional interactive objects in the programming environment.
5. Assess capabilities and limitations of the programming environment as well as the procedure used to develop the scenario-based interactive virtual environment for future end user testing.

4. FRAMEWORK OF SCENARIOS FOR EXPERIENCE-BASED DESIGN REVIEW

Tasks performed in healthcare facilities can be categorized into various scenarios, which could vary depending on the user, the type of task being performed and the issues it addresses. For instance, in a particular scenario, a nurse would receive a signal to respond to a patient call at the nurse's station. The nurse would then navigate to the defined space, and finally perform a task within that space. Another scenario could be a facility management personnel who must perform an inspection of the HVAC equipment installed. The worker navigates to the required area to identify the equipment's location and then performs a task to complete the inspection. Another way to categorize scenarios would be based on the nature of tasks being performed and defining the steps involved in them in detail. This would help identify the objects and their corresponding behaviors that will be required for the simulation of each scenario. Most importantly, this would also help determine the level of detail (LOD) required for modeling the digital content (facility, objects, avatars) in various authoring tools, to be able to simulate these scenarios effectively. For instance, a way finding scenario that requires a hospital visitor to walkthrough one end of the facility to another may not need the LOD that will be required for detailed tasks that take place in a fixed location such as a scenario where the nurse needs to check a patient‘s blood pressure

To enable the user to perform typical daily operations in the virtual environment, a framework was developed for structuring these daily tasks into scenarios. For instance, one task that may be identified is the need to quickly transport patients from the hospital entrance to a specific location within the Emergency Department (ED) for providing critical care to the patient. This scenario will have multiple forms of information that must be structured and stored to accurately allow a user who is reviewing the design to experience this task. These elements could include location and path information; information regarding temporary artifacts within the space (e.g., a patient bed, the patient, and possibly other elements that are not part of a permanent design model); and information regarding the task participants and their representation (e.g., the task may be performed by two or three individuals in a collaborative manner). This particular scenario can be represented in a formal structure that is interpreted by a computer application. Moreover, a data structure has been developed for collecting the scenario information in a format that can be referenced in the interactive virtual environment.

4.1 System Architecture of the EVPS Application

The scenario framework primarily consists of three components- the element library, the scenario engine and the user scene. The scenario engine is the main component of the EVPS application as it combines the task-based scenarios with geometric data and user input in the virtual environment. The element library is associated with other applications, databases and libraries. The user scene displays the virtual prototype and obtains feedback through the user input. The system architecture for the EVPS application is shown in Figure 1. Within the application, the 3D *Element Library* consists of three types of geometric models: the space (facility) models, the object (equipment) models and the avatar (user role) models. While the facility and equipment/ object models are stored in a *geometric database*, the user role models are retrieved from the *Avatar Object library*. The *Scenario Engine* allows for the addition of scenario and task tracking scripts by attaching various objects to these scripts from the *Scripting Library*. The *User Scene* of the EVPS application contains the 3D rendering module and a GUI widget that displays the virtual healthcare facility prototype scene along with the objects and user roles on the user's screen or output device. The objective is to allow the user to navigate through the space, interact with various objects and perform specific tasks and scenarios within the virtual facility prototype dynamically.

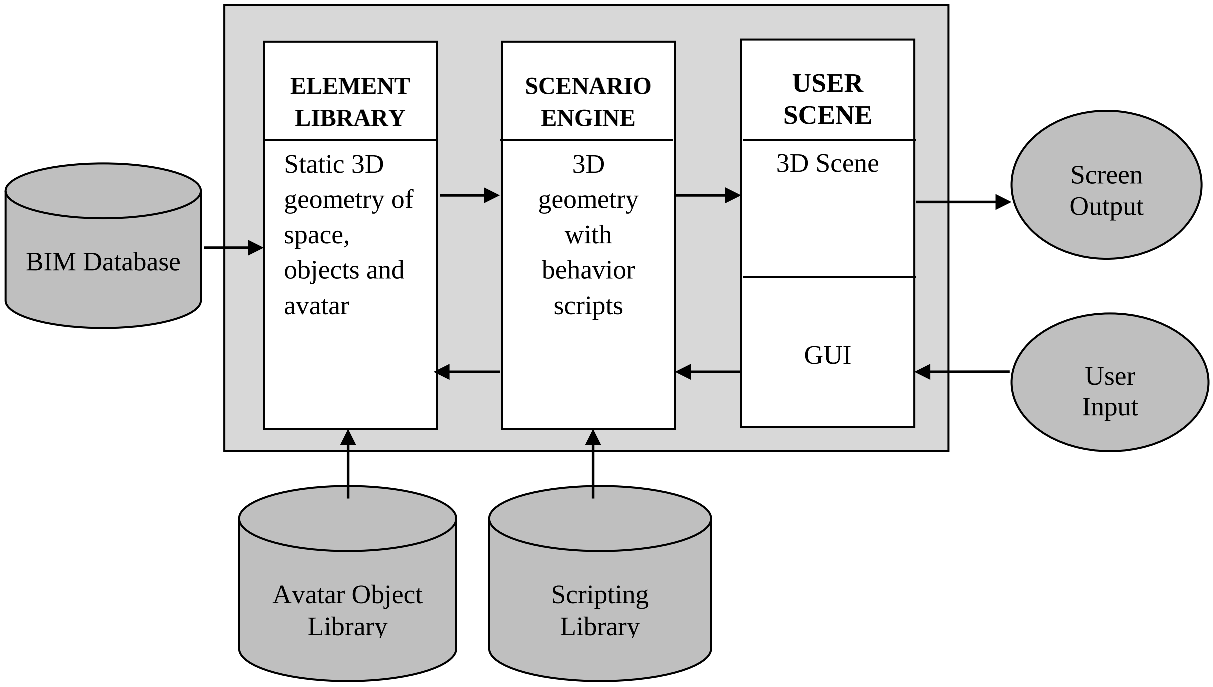

FIG. 1: Experience-based Virtual Prototyping System (EVPS) Architecture

This generic system architecture could be implemented in various game engine environments, but the specific structure of the data files and formats will vary depending on the game engine. A specific example of how separate databases and libraries are integrated using the Unity game engine is discussed in Section 5.

4.1.1 Element Library

The element library functions as a repository for 3D geometric data of the facility including healthcare equipment and other objects that are required to be displayed in the virtual environment. Since most of the 3D geometric data is authored in various BIM authoring tools, the application needs to be interoperable with these tools such that all the data is stored in file formats that are easily exported from authoring tools and imported in the EVPS application. The element library communicates with the *BIM database* to get relevant 3D geometric data of the facility that will be required for design review within the virtual environment. Apart from 3D geometric data, real-time rendering requires additional attribute information regarding texture, material and lighting. Other important attribute information required during real-time walkthroughs of a facility in a virtual environment are collision detection on the wall, floor, ceiling or other similar elements within the facility. In addition to geometric and attribute information, certain objects also have inherent behaviors that are included in the element library. For instance all the door elements in the facility could be animated to slide or swing open depending on the type of door and direction of hinge. Healthcare equipment objects such as the patient bed could be animated so that they are configured for the user (patient) to sit or lie on them. The element library combines the 3D geometric data obtained from a BIM authoring application with all the additional behaviors described above such as attributes, physics and animations associated with various elements of a healthcare facility.

In the future, it is intended that the element library will also contain information regarding the avatars of the user roles used in the scenario-based design review of healthcare facilities in the virtual environment. For the purpose of this paper, the user role is depicted as a first person controller (FPC) and hence the user or its avatar is not visible during real-time navigation of the model in the virtual environment. However, at a later stage, it is envisioned that the element library will store 3D geometric data for the avatar representation based on the user role along with behavior, features, functions and constraints associated with each type of user role.

4.1.2 Scenario Engine

The implementation of scenarios for the purpose of interactive design review in a virtual environment takes place in the scenario engine. The scenario engine links various behaviors, scenario and task tracking scripts from the scripting library to the 3D geometric objects and model in the space. The organization of this information is based on a hierarchical data structure wherein each design review *space* of the healthcare facility can be reviewed by various *user* roles as shown in figure 2. These user roles range from AEC design professionals and facility managers, to the end users such as nurses, patients and other healthcare staff. Since every user would perform a distinct task within the space and have a different agenda for design review, the functions and features afforded to the user role chosen within the virtual environment are different. For instance, a nurse may be interested in knowing if they are able to carry out certain duties like placing a particular piece of patent monitoring equipment near the patient's bed in a convenient manner. However, a facility manager may be more interested in checking the location and ease of access for various air filters that may need to be replaced in a fan coil unit. Therefore, each user role should have a different Graphical User Interface (GUI) that can be customized for that particular user. The GUI displays a distinct set of *scenarios*, which are further broken down into a series of *tasks*.

The functions performed by the scenario engine are as follows:

- Load appropriate healthcare space or facility model from element library in the user scene.
- Based on the space chosen, receive user input of the role in healthcare facility design and then load the first person controller (FPC) or avatar with relevant behavior for the user role. The GUI elements with functions and features appropriate for the user role will be loaded based on the user role selection.
- Load scenarios from a number of available scenarios defined based on the user role. Display GUI elements, additional objects needed for the scenario, and the list of tasks that are performed during the scenario.
- Once a scenario is activated, keep track of the steps or tasks performed to complete that particular scenario by updating and retrieving scenario conditions from the behavior scripting library.

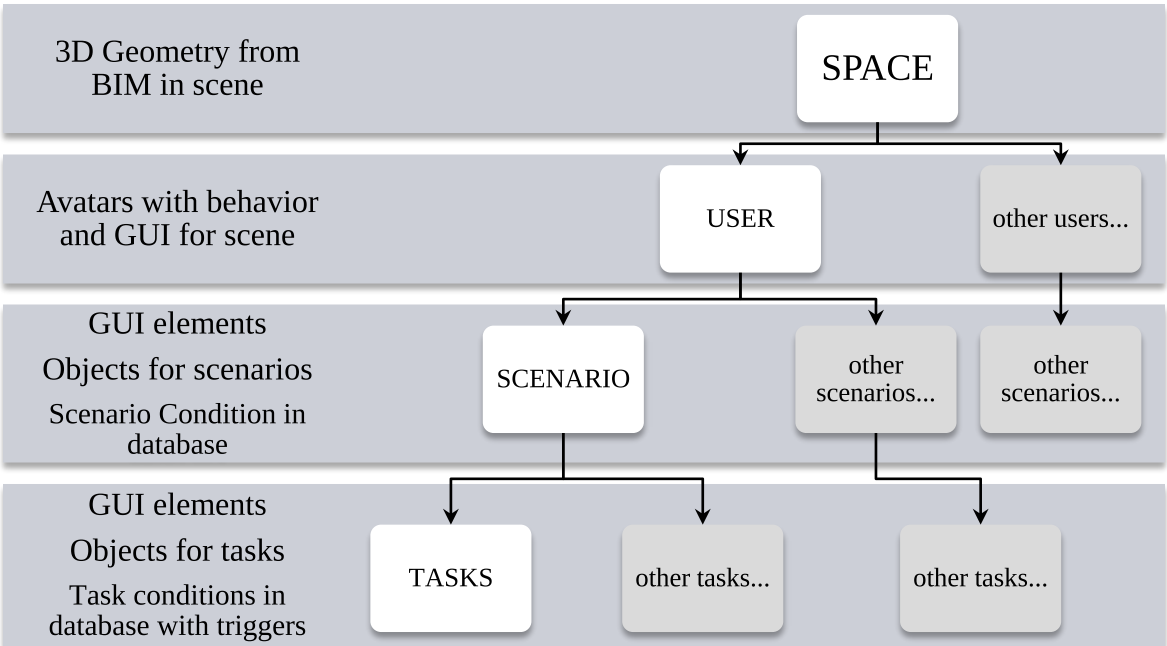

FIG. 2: Data Hierarchy within the Scenario Engine

Any given scenario contains a series of tasks to be completed in a sequence of steps. A list of instructions is displayed in the GUI for each scenario that informs the user of what needs to be done in the virtual environment. Each scenario or task within it has various objects associated to it, where each object has information pertaining to its initial and final conditions such as the orientation, position and animation required. The behavior and scripting library also contains an overall description of the nature and role of the objects used during a task or scenario. Table 1 shows an example of summary level information required for each scenario and task. In this particular scenario example with the user role of a nurse, the patient requires immediate medical attention and the nurse needs to take a crash cart to the patient bedside. Various objects, tasks and parameters used in the scenario are defined. These include starting condition of the user as well as the physical coordinates of the start and end position and orientation of the user. In this case the nurse begins the scenario in the nurse's station and ends in the patient room near the patient's bed with a crash cart. Since, each scenario consists of a series of tasks, GUI elements displaying instructions, arrows indicating objects to interact with as well as the progress status are displayed.

Furthermore, each task is associated with objects for the user to interact with in the virtual environment. The objects can either be transformed by the user during a task or function as triggers for activating other tasks. All these objects are described with similar parameters as in the case of a user role in a scenario, such as start and end conditions, positions and orientations for each object while a task is performed. These parameters help keep track of the completion of each task and therefore the status of each scenario.

Table 1: Example of information for scenarios and tasks

Scenario:

User Role:	Nurse
Description:	Patient requires medical attention... need to carry equipment to patient room
Starting Condition:	Nurse is in the nurse's station
Start Position/ Orientation:	X_{n0}, Y_{n0}, Z_{n0}/ X_{n0}°, Y_{n0}°, Z_{n0}°
End Position/ Orientation:	X_{n1}, Y_{n1}, Z_{n1}/ X_{n1}°, Y_{n1}°, Z_{n1}°
Tasks:	Take a crash cart to patient's room
GUI Elements:	Instructions, Arrows, Progress Status

Task1:

Object1:	Crash Cart
Task Type:	Move crash cart object from location A to B
Start Condition:	Crash Cart outside the patient room
End Condition:	Crash Cart next to patient bed in patient room
Start Position/ Orientation:	A: x_{c0}, y_{c0}, z_{c0}/ x_{c0}°, y_{c0}°, z_{c0}°
End Position/ Orientation:	B: x_{c1}, y_{c1}, z_{c1}/ x_{c1}°, y_{c1}°, z_{c1}°

4.1.3 User Scene

The user scene is an interface between the project stakeholders (design reviewers) and the scenario engine. The user scene comprises of two elements that are displayed to the reviewer- the navigable, real-time rendered 3D model elements and the graphical user interface (GUI) to display options to the users and get feedback from them. Based on the output screen modality, the 3D scene can be rendered on scalable output devices from a desktop machine to highly immersive virtual environments. On the other hand, to obtain feedback from the reviewer, the user scene is designed such that various input modalities can be used and interchanged depending on the type of task being performed.

5. IMPLEMENTATION STRATEGY OF VIRTUAL PROTOTYPE SYSTEM

Once the framework for the experience based virtual simulator was developed, a strategy for implementing the virtual prototype system in a real-time interactive virtual environment or 3D game engine was defined. The objective was to develop an interface for the reviewers that effectively conveyed what scenarios and tasks needed to be performed in the virtual facility. According to Karlsen (2008), the core of most massively multiplayer online role-playing games (MMORPGs) revolves around completing quests, or a specific set of actions and tasks to be performed and completed by a player for in-game rewards. Due to this nature of MMORPGs, a clear similarity exists between them and the concept for the scenario-based EVPS application environment, which this project seeks to develop. The World of Warcraft (WoW) and Everquest, two of the more commercially successful MMORPGs, have interfaces that are highly streamlined and very accessible for new users. Therefore, key interface elements of such MMORPGs greatly influenced the development of the EVPS application and several elements were adapted to better fit the needs of this project based on the design review requirements. More specifically, these elements were:

- A section of the interface to keep track of the steps in a given task along with completed, current, and future steps clearly listed on the interface.
- A region of the interface denoting the position of the user with respect to the environment in plan view, referred to as a minimap. The locations of objects that require interaction while performing the current task are highlighted on the minimap to give the user a sense of direction.
- A region of the interface that denoted the current user's name and credentials
- A feedback feature with a text input box to allow the user to leave design comments for the design professionals, and to communicate with other users in a multiuser scenario, which is planned to be added in future developments.

Functioning very similarly to MMORPG counterpart, some of the features developed for the Experience-based Virtual Prototyping Simulator (EVPS) were the project task tracker that was conceptually developed to keep track of the current step in an active scenario as shown in figure 3. Similarly, the minimap was used to display a plan view showing the position and orientation of the user within the facility while completing tasks and scenarios. During design review, it is envisioned that the minimap will help the users locate themselves within the facility. The feedback/ comment box will be incorporated in the future development to serve as a means for users to leave feedback about issues such as improperly sized doorways for equipment and poor room layout for the design team to address. This feedback would be stored in an external file and later sent to the design team. The user information and credential panel was intended to display the custom menu with a relevant list of scenarios based on the user role chosen as well as to denote areas of the hospital that their user role would have access to.

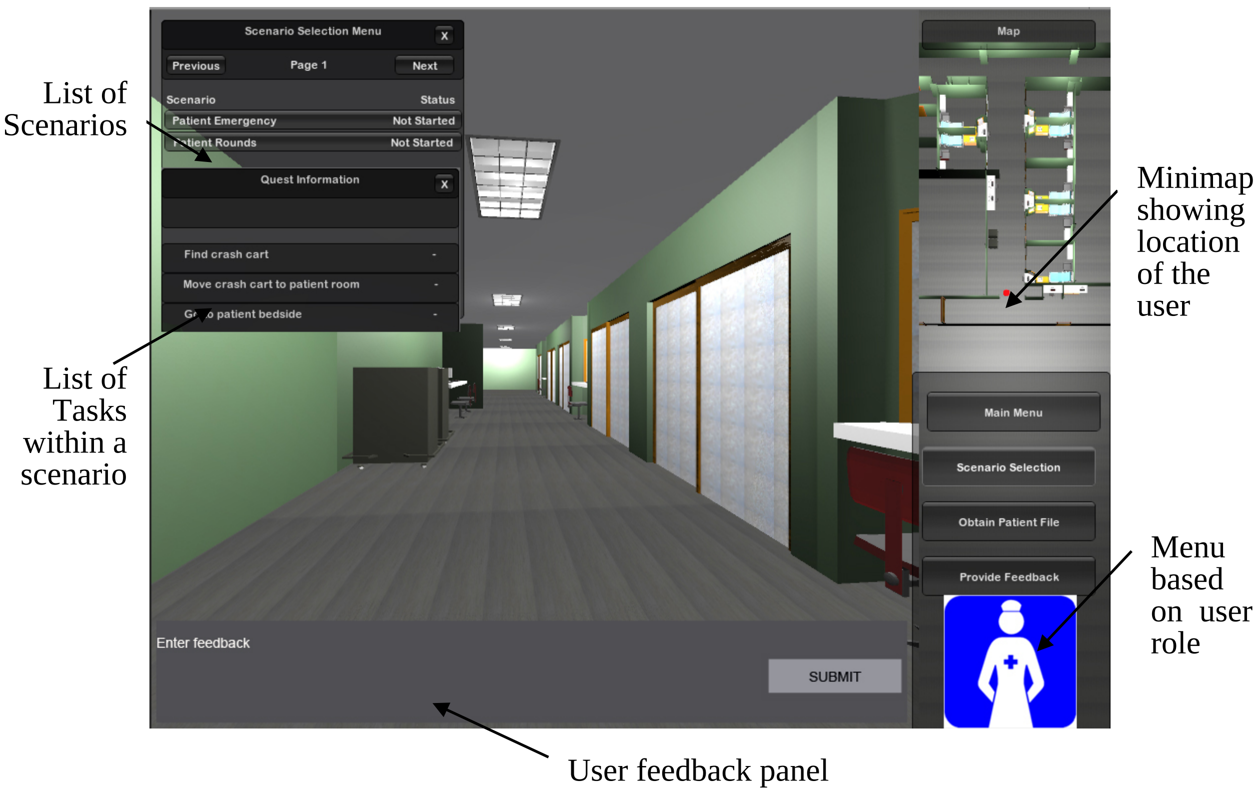

FIG. 3: Graphical User Interface of the EVPS prototype application

5.1 Programming Environment Selection: Unity Game Engine

The Unity game engine was considered a feasible option for the development of the Experience based Virtual Prototyping Simulator (EVPS) application. The Unity game engine was chosen since it has a fast rendering environment, a robust feature set that allows for customization, is affordable, and has a relatively easy interface with drag and drop ability making it easy to learn and use. The game engine can also be used in both the Mac and PC operating systems and it uses the JavaScript and Just-In Time (JIT) compilation within the C++ mono library. For the purposes of physics and rendering, it uses the Nvidia PhysX physics engine, OpenGL and DirectX for 3D rendering and OpenAL for audio. There are also various workflows for conveniently transferring geometric data from BIM and CAD authoring applications such as Autodesk Revit, Google SketchUp and Autodesk 3DS Max to the Unity Game engine. This helped us develop the level of information transferred such as textures as well as other intelligent information attached to the Building Information Model.

The EVPS application was designed such that it allowed a scenario to be loaded within the game environment. Each scenario consisted of initially, the scene or the facility depicting particular healthcare spaces. Once a specific space was chosen, the GUI consisted of various roles within the healthcare facilities which would be loaded as interactive avatars in future development. Lastly, having chosen the healthcare facility space and role, the EVPS application loaded relevant GUI components with corresponding scenario information as well as furniture, fixtures and equipment. Workflows for both importing content and applying behavior were developed for the application. One of the requirements of the EVPS application was to open and load the virtual models of the healthcare facilities such as patient rooms, operating theatres, the Emergency Department (ED) and so on. Case studies of hospitals were considered and the Autodesk Revit models were converted into FBX files to be imported into the Unity game engine.

Within each Unity project folder, there is a default Assets folder that performs the function of the element library for the EVPS application and stores all the data associated with the healthcare facility. The user scene file is located in this Assets folder and includes the main levels as well as different zones and spaces of the healthcare facility. Moreover all elements that are ever used in the user scene including game objects, prefabs (objects with attached behavior scripts), textures and other components along with their behavior scripts are stored in the assets folder. These elements, referred to as assets in the Unity game engine can be reused from one project to another. Within the Unity game engine interface, the elements stored in the assets folder are displayed in the *Projects* tab as shown in figure 4.

5.2 Developing Design Information Workflows for the EVPS Application

One of the objectives of the project was to streamline the development process so that the amount of time and effort required to create virtual prototypes of healthcare facilities was reduced. This was done through the development of design information workflows from 3D modelling software to the EVPS application being developed within the Unity engine. Various workflows were developed to best utilize 3D content from different BIM authoring tools for use in the Unity engine. The goal was to transfer as much model content as possible such as 3D meshes, textures, and lighting attributes into Unity to ensure that limited amount of modelling was needed within it. For this purpose, workflows from Autodesk Revit, 3D Studio Max, and Google SketchUp were tested.

It was found that the most successful workflow for this application was from Autodesk Revit directly into Unity through a FBX file format. The advantage of using existing BIM authoring tools, such as Revit Architecture was that it allowed the use of existing building information models for the development of the virtual prototypes in the EVPS application. Another benefit of directly exporting models into Unity was that design changes made to the model in the native authoring tool (Autodesk Revit) could simply be exported again to Unity as it would be able to update those changes. However, during workflow development, the design team also encountered certain interoperability issues while importing models from Revit into Unity which were primarily related to textures, lighting, and overall organization of the model hierarchy. A screen shot of the Unity game engine is shown in Figure 4.

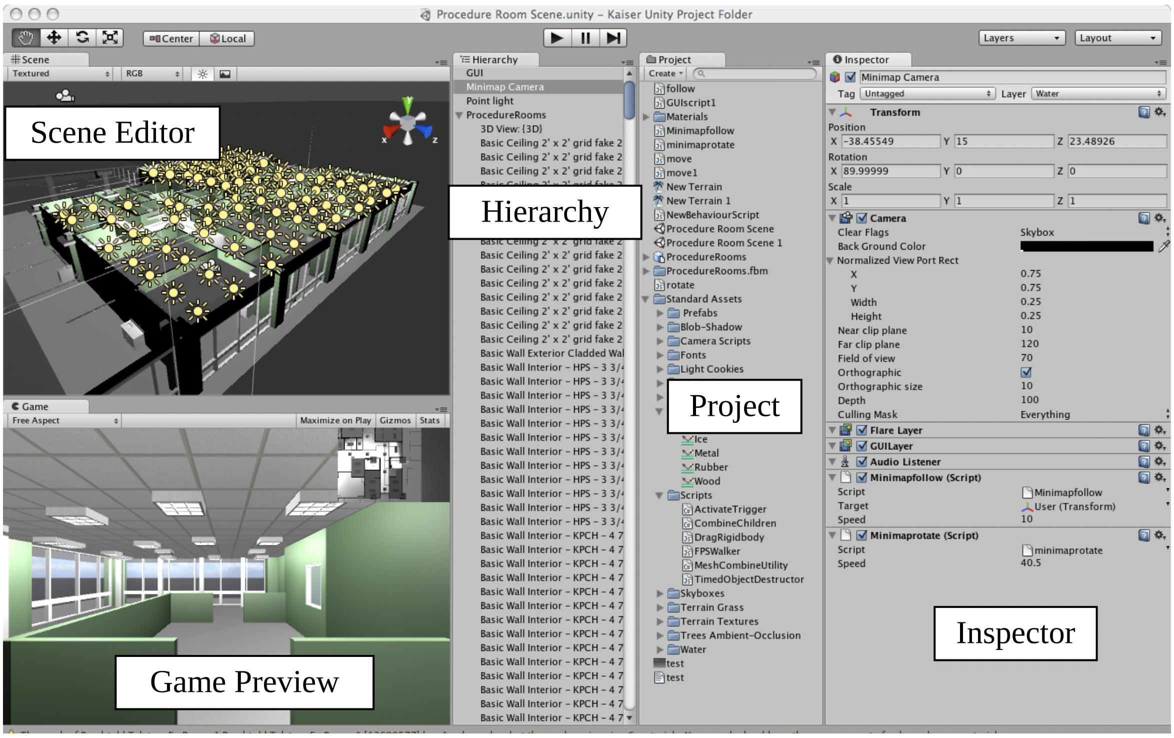

FIG. 4: Unity interface showing the applied texture and lighting as well as the information structure of a healthcare facility prototype.

5.2.1.1 Materials and Textures

The healthcare facility models were exported from Autodesk Revit in an FBX file format that could easily be imported into Unity. One of the benefits of using a FBX file format was that when files were transferred within the Autodesk suite, it retained information pertaining to 3D meshes, texture and camera locations. However, when a FBX file format was exported into Unity, it lost any type of textural information associated with the model and the materials and textures had to be reapplied. Since the Unity engine had an extensive in-built library of textures and material, the design team was able to apply basic realistic textures within Unity with relative ease, but it certainly required additional effort. Other workflows, particularly through other visualization applications, may not encounter this texture issue, although it has not been uncommon for us to have similar issues when moving from CAD / BIM authoring applications to interactive game engines.

5.2.1.2 Lighting

While 3D meshes of the lighting fixtures did transfer successfully into Unity, the lighting characteristics did not. The design team had to add the lights for the spaces within the Unity engine. Typically, a large amount of time is required in making the lighting as realistic as possible; hence successfully carrying through these characteristics can translate into potential time savings for the project team. One possible solution to this lighting issue could be incorporating the use of baked textures within the lighting workflow. By importing a facility model into 3D Studio Max and using render to texture for the objects, baked textures depicting the lighting effect can be transferred to the Unity model. However, this may also raise the issue of redoing this entire process whenever revisions are made to the design of the model in Revit before being transferred again to Unity.

5.2.1.3 Data Structure

Another important lesson learned during the workflow development was the efficient organization of objects within a facility model based on various behavior categories could greatly help in expediting the scripting process. This is because one cannot sort model objects into groups based on either similar attributes or their common location within the Unity game engine. To be able to manage larger models in Unity, it was important to have a hierarchy of objects within the model so that scripts could be attached to multiple objects that had similar attributes at the same time. For instance, by grouping all wall objects together, collision behavior could be attached to the entire wall group at the same time rather than attaching it individually to each wall element. It was found that by setting up multiple “exporting” views within Revit and sorting the model by each of the object types (i.e., walls, ceiling, lights, etc.), grouped objects could be imported into Unity separately. These groupings of objects could then be setup within each view followed by adding scenarios as required. Moreover, scripts could also then be attached to an entire group of objects, hence speeding up the development process. Future plans for more selecting searching and filtering of objects inside the game engine authoring environment will help alleviate some of the challenges related to grouping.

5.3 Developing Behavior Scripts for the EVPS Application

After the 3D geometry was transferred to the Unity engine, textures were applied to all the elements and lighting was added, the next step was to add behaviors comprising of physics, collisions, and triggers. The physics and collision behaviors applied to the 3D geometric elements in the virtual facility were either static or dynamic. While static behaviors were applied to immoveable objects; dynamic behaviors were applied to moveable objects and user roles. For instance, while colliders were applied to walls and objects in the scene (space), gravity was applied to the First Person Controller (FPC) as well as other moveable objects. Similarly, in the case of doors, either collider behavior was not attached to them so that the FPC could walk through them or they were animated to open whenever the FPC was in proximity. Another form of behaviors applied to the interactive objects within a model related to the role these objects played in various tasks and scenarios within the facility. Examples comprised of objects such as trolleys, carts and patient beds that the FPC needed to move or interact with based on the scenarios or tasks being performed within the application. However, applying behavior to objects that were a part of a task or a scenario was more complex and required scripting of “triggers” that conveyed interaction of the FPC with the objects as well as an overarching script to monitor and count the steps, tasks and the overall scenarios being performed in the virtual prototype.

5.3.1 Scenario Tracking Process

To manage the scenario tracking process, two scripts with unique roles were developed- the *Interface Log Script* and the *Object Trigger Script*. The Object Trigger Script was responsible for interpreting whether the user had interacted with an object (e.g., a door or piece of equipment) in the correct order with respect to the overall scenario. A different object trigger script had to be assigned to every interactive game object to accommodate for different user interaction types; direct physical manipulation and user proximity were the two most common interaction types.

The Interface Log Script was written in conjunction with the main user interface itself. The purpose of this script was to monitor changes to global variables generated from the Object Trigger Scripts and update the task information in the user interface accordingly when a task was performed. This flow of information is displayed graphically in Figure 5.

Several obstacles currently exist which prevent total automation in the scripting process. Foremost, each of the Object Trigger Scripts had to be configured individually with respect to what actions activate a trigger (e.g., proximity, physical contact, user selection). Since each interactive game object was assigned a unique identity, the object trigger script was applied to each object via a simple drag and drop interface in Unity. In the future, applying the trigger script to each individual game object would greatly increase the amount of code and time required to develop the various scenarios within the application. Hence, it would be essential to first develop an organized data structure where each interactive object belongs to a specific family of objects that have similar interaction behaviors. After structuring this data, advanced coding techniques and programming strategies need to be implemented that allocate certain template scripts to a corresponding family of objects. As an example, all objects that belong to a door family could have a swing animation script attached to them and a trigger script that enables them to open in close proximity to a reviewer’s FPC.

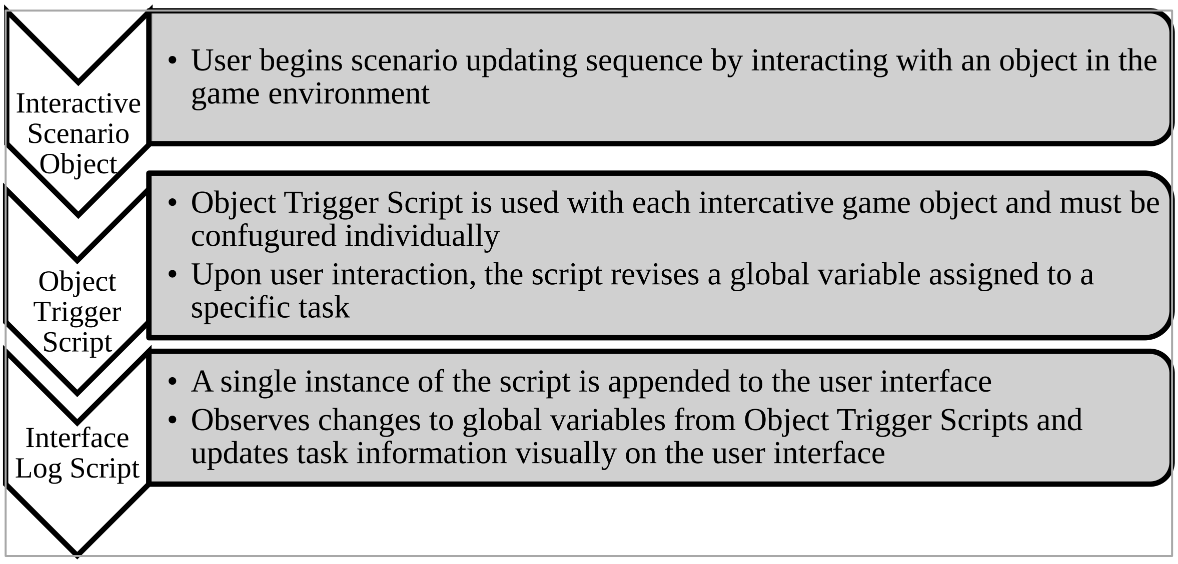

FIG. 5: Scenario scripting steps

5.4 EVPS Application Prototype

The Experience based Virtual Prototyping Simulator (EVPS) is still under development and screenshots of the proof-of-concept can be seen in Figures 6 to 11. As soon as the application is launched, a start menu is displayed that welcomes the user to the EVPS application. The user can then choose the type of room or facility they want to review by selecting the images of the rooms, which would then be loaded in the game engine. After having chosen the room, the user is asked to choose a particular role within the space (patient, nurse, and facility manager) based on the room chosen. Once the user has made these choices, the EVPS application loads the chosen facility, user role along with the GUI, menu items and the corresponding scenario quests after which point the user can begin the design review.

Once inside the virtual model, the user is able to navigate the model via a First Person Controller (FPC). Based on the user role chosen, they have custom settings and they are able to undertake relevant scenarios, each with a list of tasks that could include retrieving information, interacting with objects and elements in the facility, and also being able to alter certain object locations. A common feature across all user roles is the minimap which is located in the top right corner and serves as a reference for way finding and location tracking during walkthroughs, giving a better understanding of the user's location within the facility.

The following figures show a particular scenario that requires a nurse in the patient room. The scenario requires the nurse to go to the nurse's station, receive scenario instructions and a list of three tasks. These tasks include locating a crash cart, taking the cart to a patient's room and then navigating to the patient's bedside. The application keeps track of each task that is completed and once all tasks are completed, it displays that the scenario has been completed. A video of the EVPS with this scenario can be also be seen by clicking on the following link.

FIG. 6: Start menu of EVPS application prompting user to choose a space

(Video link: http://dl.dropbox.com/u/2459431/Unity%20EVPS%20project%206_23.wmv)

FIG. 7: Start menu of EVPS application prompting user to choose a specific user role

FIG. 8: Scenario selection menu and GUI customized for the nurse role

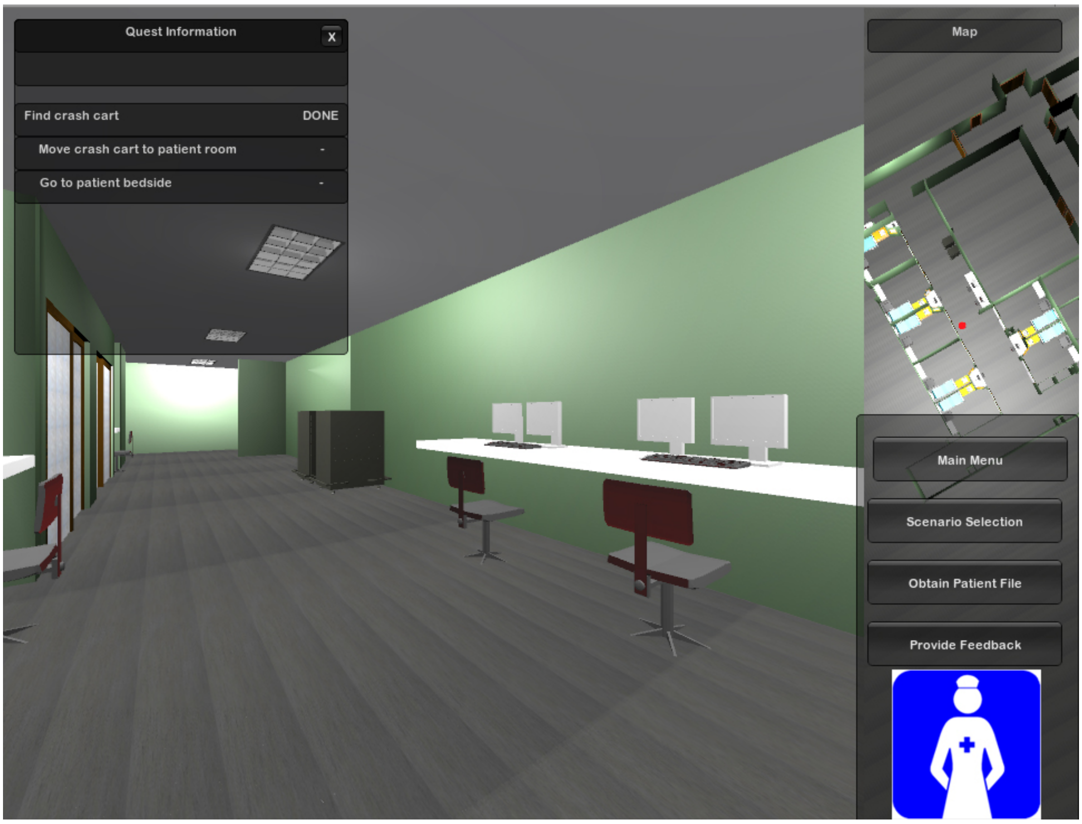

FIG. 9: List of tasks in a particular scenario displaying that the 1st task has been completed

FIG. 10: List of tasks in a particular scenario displaying that the 2nd task has been completed

FIG. 11: Menu displaying that the 3rd task and an entire scenario has been completed

6. CONCLUSIONS

This paper illustrated the use of game engines for effectively developing interactive virtual facility prototypes for the purpose of experience based design reviews, by enabling simulation of scenarios that require specific tasks to be performed, within the virtual environment. First, the paper presented a framework for the development of interactive experience-based virtual prototypes for reviewing designs. This framework consists of three components for incorporating task-based scenarios that take place in a healthcare setting. These are the element library, the scenario engine and the user scene. While the element library is the repository of all the data and content required for the healthcare facility prototype, the user scene is the interface between the reviewer and the facility content. The most important component that bridges between the element library and the user scene is the scenario engine. It performs the function of linking relevant scripting behavior to interactive objects as well as keeping track of the tasks being performed in the scenarios that are based on the user role chosen by the reviewer.

This framework was implemented through the development of a prototype application for reviewing the design of hospital facilities termed as the Experience based Virtual Prototyping Simulator (EVPS). The Unity game engine was chosen for the development of the EVPS application due to its fast rendering environment, robust feature set and user friendly interface. Design information workflows were developed to transfer 3D geometric content of the healthcare facilities from various BIM authoring tools to the Unity game engine. Some interoperability challenges were encountered that led to important lessons learned for future workflow developments. Issues included difficulty in transferring textural information and lighting effects along with the need for better data organization within the authoring tools prior to importing the model into the Unity game engine. Next, the development of behaviour scripts was discussed and the strategy for keeping track of scenarios and the steps or tasks being performed within them was developed. Finally, the under development EVPS application was presented by showing step by step screenshots of a particular scenario that involves a nurse to move a crash cart inside a patient's room.

6.1 Challenges with Scenario Development

One of the key challenges that became apparent during the EVPS application development is the need to investigate better methods of interoperating between BIM and other geometric authoring tools and game engines. For the development of the prototype system, a workflow between Autodesk Revit and Unity allowed for transfer of geometric data into the EVPS application. While geometric information pertaining to the virtual models has been relatively easy to transfer between different applications, there is a need to address information which is necessary to develop more interactive scenarios. For example, behavioural attributes or characteristics for objects within the facility do not exist, and this interactivity currently must be entered for individual components, e.g., scripting animation for the automatic opening of a door. In the future, more intelligent ties between static geometric components in design applications need to be developed to allow for interactivity with the objects within dynamic game engine environments. This is also true for property information such as weight and friction that are needed to accurately represent the physical properties of an object in a game engine.

Another observation from the AEC design review perspective is that certain enhancements to present game engines could be very beneficial for the efficient development and implementation of applications such as the EVPS. For instance having "plug n play" components for features like minimaps, feedback or comment boxes, and measuring tool in the game engines would limit the need for each organization to develop these elements. Moreover, the addition of other functionality such as clipping planes and the ability for selective viewing of components within the virtual models could be extremely useful during design review of virtual prototypes. These are standard features within CAD design applications, but game engines are functionally developed to be immersed within a 3D environment so these additional selective viewing functions must be programmed into the applications. Having these functionality and features built into the game engines would greatly save on the coding and scripting required to custom create some of these features for every project.

In addition to the technical implementation challenges, there is a need to document job tasks through ethnographic research on what people do to scenarios. Observations should focus on the different types of tasks healthcare staff members carry out along with how the physical environment supports or hinders their ability to carry out these tasks. Ethnographic research studies (Strauss, 1987; Reddy et al., 2002) have defined procedures for data collection and analysis that can accurately document tasks performed in healthcare facilities. An important next step is to determine how to use these methods to develop scripted task-based scenarios that can be used by the EVPS application.

6.2 Future Research Direction

The model content used in the EVPS application has been developed from an actual healthcare facility project. The next step will be to assess the capabilities and limitations of the developed EVPS application through user-testing during the design review sessions by involving AEC professionals and end users of this project. This would provide added insight into the value of performing the experience based design reviews in virtual environments. Based on the results and feedback from the user-testing of the application, revisions will be made and additional functionality will be developed.

Since the greatest use of this application will most likely occur early on in the design process, it will be important to further explore alternate workflows that incorporate interoperability of design information from conceptual design tools such as Google SketchUp. Another requirement is to create digital model content of various healthcare equipment and furniture within the game engine. The digital model content developed could comprise of patient beds, crash carts, trolleys and other healthcare related furniture. The goal of this future research project will be to create and accumulate over time an open source library of reusable digital model content of equipment that can be used in many healthcare facility design projects. The development of this readily available content could be instrumental in making the design information workflows more achievable in a timely and productive manner.

As part of the future continuing goals of this project, it is planned that more features and functionality will be added to the developing EVPS application. During the design and development of features, the user interface will be assessed based on design review requirements. These features could include tying the feedback and comment box within the EVPS application to a BIM database that would keep track of design and intent changes along with the source of design change information. According to the design criteria, other features and functions planned to be incorporated in the EVPS application include a move tool to move objects around the facility, an inspect tool to retrieve information about various objects and a measure tool to calculate distances between objects and various points.

Another future requirement is to add various stakeholders and end users of healthcare facilities as 3D avatars or characters of the nurse, patient and facility management personnel. These avatars will also include behavioural characteristics based on each user role to make the simulation of scenarios and performing tasks seem more real. This would enable both designers and end-users to explore the virtual prototype through either one of the roles and help them gain insight from various perspectives during the design review process. Another scope of work for the future is to investigate incorporating a diverse range of input modalities such as joysticks and touch screens to enhance the interactivity and scalability of the EVPS application.

Acknowledgements

We wish to thank the National Science Foundation for their support of this research (Grant No. 0348457). Any opinions, findings, conclusions, or recommendations are those of the authors and do not reflect those of the National Science Foundation or the project participants.

7. REFERENCES

Bate, P. & Robert, G. (2007) *Bringing User Experience to Healthcare Improvement: The Concepts, Methods and Practices of Experience-Based Design*. Oxford, U.K., Radcliffe.

Bate, P. & Robert, G. (2006) Experience-based design: from redesigning the system around the patient to co-designing services with the patient. *Quality and Safety in Health Care*, 15, pp.307-310.

Dunston, P.S., Arns, L.L. & McGlothin, J.D. (2007) An Immersive Virtual Reality Mock-Up for Design Review of Hospital Patient Rooms. In: *CONVR 2007*. Penn State University, University Park, PA, http://www.engr.psu.edu/convr/proceedings/papers/01_Dunston_submission_45.pdf, p.9.

Eberly, D. (2007) *3D Game Engine Design: A Practical Approach to Real-time Computer Graphics*. 2nd ed. San Francisco CA ; Oxford, Morgan Kaufmann; Elsevier Science.

Fritsch, D. & Kada, M. (2004) Visualization Using Game Engines. In: *ISPRS Commission 5*. Istanbul, Turkey, pp.621-625.

Hamilton, D.K. & Watkins, D.H. (2009) *Evidence-based design for multiple building types*. Hoboken N.J., Wiley.

Karlsen, F. (2008) Quests in Context: A Comparative Analysis of Discworld and World of Warcraft. *International Journal of Computer Game Research*, 8 (1). Available from: <http://webcache.googleusercontent.com/u/gamestudies?q=cache:ZWE8phf-T1IJ:gamestudies.org/0801/articles/karlsen+quest&cd=1&hl=en&ct=clnk&gl=us&ie=UTF-8> [Accessed 28 June 2010].

Leicht, R.M. & Messner, J.I. (2009) *Comparing physical and virtual mock-ups: A case study*. University Park, PA, USA, The Pennsylvania State University.

Luck, R. (2003) *Dialogue in Participatory Design*. Elsevier Publishers, Ltd.

Majumdar, T., Fischer, M.A. & Schwegler (2006) Conceptual design review with a virtual reality mock-up Model. In: Montreal, Canada.

Maldovan, K.D. & Messner, J.I. (2006) Determining the effects of immersive environments on decision making in the AEC industry. In: *Proceedings of the ASCE Joint International Conference on Decision Making in Civil and Building Engineering,* Montreal, Canada.

Manning, R. & Messner, J.I. (2008) Case studies in BIM implementation for programming of healthcare facilities. *Journal of Information Technology in Construction,* 13, p.p. 246.

Norman, D. (1988) *The Design of Everyday Things*. New York, Doubleday.

Nutter, E. (1995) Participatory design practices: a special interest Group. In: *Human Factors in Computing Systems*.

Reddy, M., Pratt, W., Dourish, P. & Shabot, M. (2002) Asking questions: Information needs in a surgical intensive care unit. In: *Proceedings of American Medical Informatics Association Fall Symposium (AMIA'02)*. San Antonio, TX.

Shiratuddin, M. (2008) *Virtual Architecture: Modeling and Creation of Real-time 3D Interactive Worlds*. Raleigh N.C., Lulu.

Stankos, M. (2007) Evidence-based design in healthcare: A theoretical dilemma. *Interdisciplinary Design and Research e-Journal : Design and Health*, 1 (1).

Strauss, A.L. (1987) *Qualitative Analysis for Social Scientists*. New York, Cambridge University Press.

Ulrich, R., Quan, X., Zimring, C., Joseph, A. & Choudhary, R. (2004) *The Role of the Physical Environment in the Hospital of the 21st Century: A Once-in-a-Lifetime Opportunity*. Funded by the Robert Wood Johnson Foundation.

Zimmerman, A. & Martin, M. (2001) Post-occupancy evaluation: benefits and barriers. *Building Research and Information*, 29 (2), pp.168-174.