8
3D NAVIGATION BASED ON 3D GAME ENGINE – Preliminary Results

Khairul Hafiz Sharkawi
Alias Abdul Rahman

Department of Geoinformatics,
Faculty of Geoinformation Science and Engineering,
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.

ABSTRACT

With new technology of hardware and software available today, more and more research on 3D capabilities were carried out, tested and incorporated with GIS as an effort to realize the components of 3D GIS such as 3D navigation. Many navigation systems use 2D maps to represent the real world objects and situations; and most of them were not able to provide real spatial information (shapes, colors or materials) of the objects or scenes, as we perceived. Features and navigational landmarks were not easily available for navigating in the real world, as they were very useful and helpful for the navigation purposes. This chapter describes an experiment on 3D navigation (indoor and outdoor) within building environment. We utilize a freely available 3D game engine as a navigation tool couple with simple programming effort. We also incorporated the navigation system with spatial database management system (DBMS). The navigation system works and will be presented. The outlook and future works on the proposed navigation system for 3D GIS will be highlighted.

1.0 INTRODUCTION

Rapid development in computer-related industries and technologies has lead to the emergence of better hardware and software that can support more sophisticated functions, smart applications and more complex (3D) visualizations and capabilities.

The need for 3D capabilities to be incorporated in computer applications has long been sought and worked on; and its demands are growing due the availability of these technologies. In GIS, the need for 3D is pushed by the increasing demand for 3D information and technologies development such as improvement of 3D data collection techniques i.e. aerial and close range photogrammetry, airborne or ground based laser scanning, surveying and GPS due to the availability of faster and more accurate sensors (Stoter et al, 2006). High quality visualization such as 3D visualization alone served as an added advantage for displaying geospatial data and information which means other 3D functionalities especially for analyses will be huge enhancements for GIS related applications. According to Coors et al. (2006), the navigational value of a 3D map increases due to the high visual correspondence between map objects and real world objects.

In 3D visualization aspect, the producers and creators of animated movies and computer games plays the leading role (Fritsch, 2003) since they have improved so much in making the visualization of 3D models as realistic as possible. The computer game industry has come a long way since they first started where they have been developing 3D game engines that are not only able to provide computer games with 3D visualizations but also other 3D functionalities. Nowadays, 3D game engines are more versatile in its implementations that make it possible to use the same packages in developing other computer applications. Fritsch and Kada (2004) stated that 3D game engines might be the missing part that enables 3D visualization in geo-related applications.

1.1 The Need for 3D in Navigation Systems

3D navigation is one of the aspects or components in 3D GIS. Most of the available navigation programs use primarily 2D plans for visualization and communication features (Meijers et al., 2004) but users are craved for more realistic visualization and functions (3D) in order for them to understand more about the environment around them i.e. landmarks, when they navigate in certain areas.

Although the current 2D navigation systems are still useable, they are starting to show some weaknesses in certain aspects. The inability to represent spatial objects accurately made it difficult for the users to relate the 3D virtual environment with the real world which might be useful for them while navigate in the real world. Coors et al. (2004) suggested that the high visual (3D) correspondence between map object and real world objects allows the user to recognize building easily.

The remainder of this chapter discusses the overview on the development of the 3D navigation system for indoor and outdoor environments using an important component in the game industry, which is the 3D game engine. It also discusses about the 3D buildings generation and the database development involved in this research work.

2.0 RELATED WORKS ON INDOOR AND OUTDOOR NAVIGATION

There are several researches that focus on 3D navigation such as Karas. et. al (2006). They have created a program that can automatically extract geometry and network model from a building's blueprint and then navigate users from point A to point B inside the building using the shortest path. Meijers et al. (2004) discussed about the 3D geoinformation indoor that focused on the structuring for evacuation using the network analysis technique. Coors et al. (2004) evaluated several means of presenting the route instruction to mobile users such as combining the 2D maps with 3D maps along with text instructions. Desney et al. (2001) have experimented a work on the alternatives view and way in exploring using 3D navigation which is the combined speed-coupled flying with orbiting while Tsai et al. (2000) investigates on the user interface in 3D navigation in order to made it easy and convenient for the users to navigate inside a virtual environment.

Some researchers that are using existing 3D game engine for GIS purposes are such as Fritsch and Kada (2004). In their research, they used several different 3D game engines such as *Quake II* (developed by *Id Software*) engine for indoor visualization, *Unreal 2* engine (developed by *Epic Games*) for outdoor visualization and *Torque* game engine for indoor and outdoor visualization. They also discussed several capabilities of these game engines that can be use to improve the 3D visualization for various fields of applications such as its capabilities to support single-user and multi-user in a network environment i.e. on LAN or internet. An on going research by Ujang et al. (2007) is working on implementing Djikstra algorithm in 3D world. They implement a modified Djikstra algorithm to find a shortest path in a 3D world rendered using 3DSTATE game engine.

In this research work, we combined a crucial component in the entertainment industry (3D engine) with GIS components in order to develop a highly optimized navigation system in GIS for various purposes.

3.0 THE 3D GAME ENGINE

A 3D engine or game engine is the core software component of an application that uses real-time graphics such as computer game and other interactive application that contains 2D or 3D graphics. The basics functionality that commonly provided along with 3D engine's SDK (System Development Kit) are the so-called renderer (rendering engine) for 2D or 3D graphics rendering, a physics engine which will handle most of the physics apply in the applications, a media engine that enables sound and video to be use in the application, scripting, animation, AI (Artificial Intelligence) and networking capabilities.

The 3D game engine used for this project is the *Truevision3D* (TV3D) engine. It is a highly optimized 3D engine that is developed based on the DirectX library platform from Microsoft. It supports multiple programming languages for developing 3D applications and games such as Visual Basic 6, Visual Basic.Net (VB.Net), Delphi, C++, C#, and any other COM-compliant language with DX library support. It is an inexpensive 3D engine that possessed features and performances of expensive high-end 3D engine in the market. The 3D engine fully supports the DirectX 8.1 and DirectX 9.0 features and several formats for 3D model (meshes) such as 3DS, X, MDL, MD2, MD3 and Skinned Mesh Support.

The TV3D engine also consists of the Media Engine, which deals with the sound and video to enhance the 3D applications. The Media Engine supports the DirectSound, DirectMusic and DirectShow functions, hardware/software sound mixing, and unlimited simultaneous sound. It also supports most of the major audio format such as mp3, WAV, MID and MOD. The full features for the Truevision3D engine can be view at www.truevision3d.com.

There are other 3D engines that can be use such as OGRE3D, Genesis3D, Irrlicht Engine, Panda3D and 3D State.

4.0 THE 3D RECONSTRUCTION OF THE BUILDINGS

The 3D buildings involved in this research are generated based on the digitized floor plans. The digitized floor plans (in AutoCAD Drawing (.dwg) format) are then imported into the 3D Studio Max software before the 3D building can be generated. Every layer that has been digitized in the floor plan is then converted into suitable layers in 3D Studio Max. The outlines of buildings and its details then will be extruded to suitable heights based on the plans. All major details of the buildings are generated according to the floor plans. However, some simplification and generalization are applied to the 3D buildings to compensate the engine's limitation.

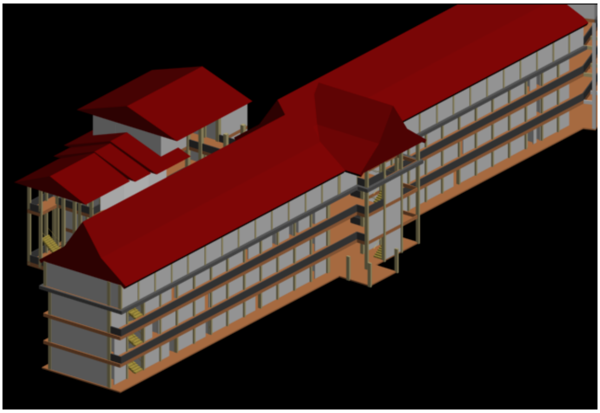


Figure 1: One of the 3D building rendered in 3D Studio Max without its textures.

4.1 Texturing the 3D Objects

All textures of the buildings involved are collected using a standard digital camera. The textures are captured section by section and it needs to be rectified before it can be applied onto the 3D models (buildings). Photomodeler Pro 5 software is used to produce ortho-projected textures using the “Rectangle on photo given by the 3 points” technique provided in the software. The corrected textures (Figure 2 b) are then joined or stitched together using Adobe Photoshop CS as showm in Figure 3 below.

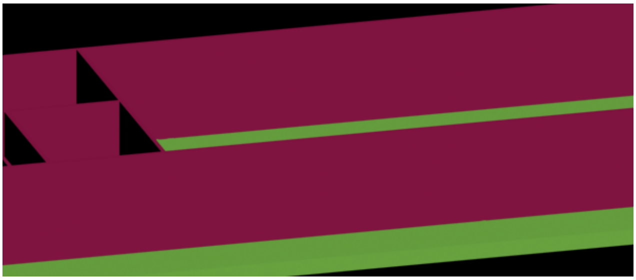
a) Original Image from digital camera

b) Rectified image

Figure 2: Photo rectifications

Figure 3: Part of the joint rectified photos

The rectified photos/textures are then applied onto the 3D buildings using the Material Editor function in 3D Studio Max software.



a) Part of a 3D building without textures

b) 3D building with textures

Figure 4: Before (a) and after (b) the textures are applied onto the 3D building.

5.0 DATABASE DEVELOPMENT

The database is developed to store all attributes data that are relevant for analyses in this project. It is designed to enable users to retrieve data based on the analyses run within the application. This attributes database will be linked with its spatial objects during the system runtime.

5.1 The Tables

The attributes database contains four major tables to store all related attributes data which are BUILDING, ROOM, OCCUPANT and NODE.

The BUILDING table is used to store data for building general information such as the building's name, type (e.g. hall, faculty, etc) and its functions such as convention hall or multi-purpose hall. Since a building may consists of many rooms, this table is connected to the ROOM table which is use to store the information of the rooms located inside a building.

The ROOM table consists of six fields which is ROOM_NO for the room number, BLOCK for the block where the room is located, FACULTY for the faculty's name, ROOM_TYPE is for the type of the room (e.g. lecture hall, tutorial room, etc), CAPACITY for the number of people can be in the room at one time, and PHONE_NO for the room's phone number (if any).

The OCCUPANT table holds information on the occupants of the rooms in a building. This table contains 5 fields which is the NAME for the occupant's name, TITLE for the occupant's title, POSITION for the occupant's position, DEPT for the occupant's department and ROOM_NO for the room number occupied by the occupant.

Since each room may have zero or more occupants while each occupant can reside in one or more room, the 'many-to-many' relationship connected the two tables. This relationship needs to be resolved by introducing a junction table derived from these two tables called OCCUPIEDROOM table. This table contains the foreign keys from both original tables as shown in figure 5.

All buildings and rooms contain possible paths or destination points that will be used in the network analysis. The nodes in these paths or destination points needs to be stored in the NODE table which contains five fields which are INDEX for indexing (this value are given automatically by the DBMS), X_COORD, Y_COORD and Z_COORD are for the X, Y and Z coordinate values respectively. DEST_NAME are use to store the room number if the nodes are on specific places.

Table 1: Sets of tables and its fields contains in the database.

BUILDING
BLOCK [P.K]
NAME
TYPE
FUNCTION

a) Table 1

ROOM
ROOM_NO [P.K]
ROOM_NAME
ROOM_TYPE
BLOCK
BUILD_NAME
PHONE_NO

b) Table 2

OCCUPANT
EMP_NO [P.K]
EMP_NAME
EMP_POS
DEPT_NAME

c) Table 3

NODE
INDEX [P.K]
X_COORD
Y_COORD
Z_COORD
DEST_NAME

d) Table 4

5.2 The ER-Diagram

The relationships between the tables are important in developing the conceptual model for the database. Figure 5 shows the Entity-Relational (ER) diagram that depicts the relationships between the tables involved.

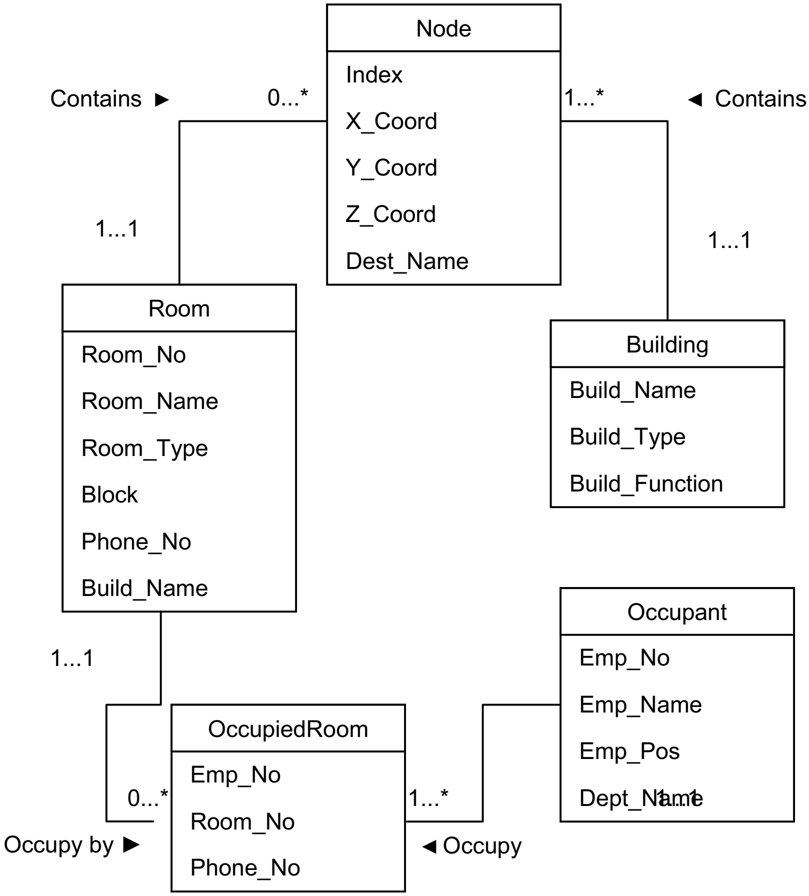


Figure 5: The ER-Diagram for the attributes database.

6.0 PATH FINDING FUNCTION FOR THE 3D NAVIGATION SYSTEM

Artificial Intelligence (AI) is commonly view as the ability of a machine, software or virtual characters to think and react like human beings. Games' artificial intelligence also can be refers to techniques used by the virtual characters to react and interact with the human characters. Path finding algorithms uses in game industry are one of the important components in the engine's Artificial Intelligence (AI). This function usually used to move Non-Player Characters (NPCs) or Computer-controlled characters to certain locations on the game's map such as towards enemy's unit or gold mines.

Djikstra's algorithm is a path finding algorithm that guarantees to return a shortest path as its result. It works by visiting vertices in the graph from the object's starting point and then repeatedly examines the closest not-yet-examined vertex, adding it to the set of vertices to be examined. Basically, this algorithm expands outwards equally in every directions (Lester, 2005) from the starting point until it reaches the goal.

The A* algorithm is a common and widely used path finding algorithm in computer games industry (Grzyb, 2005). The algorithm work similarly to the Djikstra's algorithm but it used the heuristics function (estimation value) to guide its way towards the destination in a short period of time. This will avoid it to search for the path in unnecessary directions but still return the shortest path to the destination. On the other hand, Djikstra's algorithm will end up exploring a much larger area before the destination is found which makes it slower than the A* algorithm.

In order to optimize the engine's functions, this research used the engine's built-in path finding algorithm. This engine uses the A* algorithm, which is a popular algorithm and commonly used in game industries because it is faster than any other major algorithm. The nodes for the paths network are combinations of static nodes (fixed nodes) and dynamic nodes (nodes that can be added or deletes by users at any time). The static nodes marks the common paths existed currently in the real world. Users can add new nodes (dynamic nodes) to create new paths to the existing paths if any changes occur in the future. These nodes can be added in real time when the application is running. The engine's AI will find the shortest path to the selected destination and if the path is blocked, the AI will find another alternative path.

7.0 PRELIMINARY RESULTS

From the current progress of this research, we have successfully come out with some preliminary results. The developed 3D navigation system allows users to move freely in indoor and outdoor environment in the virtual world. It also able to navigate users from point A to point B based on the path network available in the 3D world (See Figure 7). This system will find the shortest path to reach the destination and if somehow the path is blocked or become inaccessible, it will find an alternative path. The path network in the 3D world may consist of static nodes (fixed nodes) and dynamic nodes. Users can add new nodes to the static nodes to create new paths in case there is changes occur in the future. Figure 6 shows one of the buildings rendered/load into the 3D game engine.

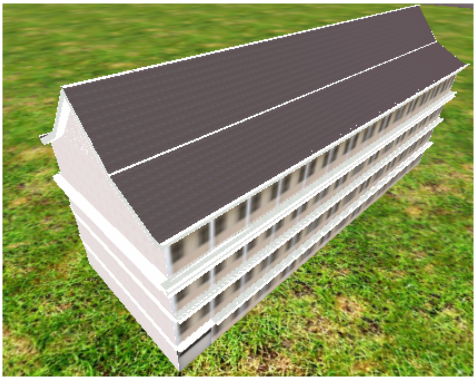


Figure 6: A building rendered in the 3D engine

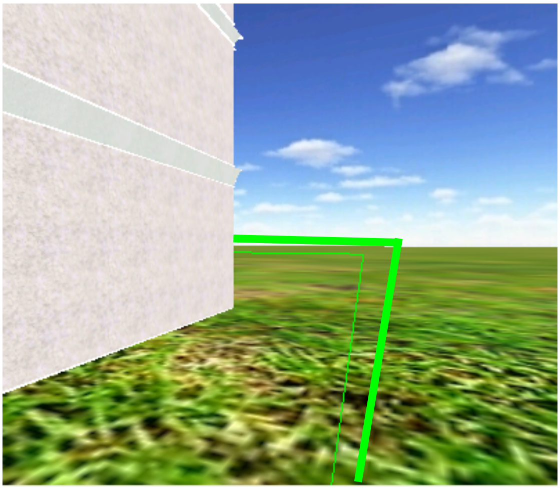


Figure 7: The system navigates users along the shortest path

8.0 CONCLUDING REMARKS AND FUTURE WORKS

This paper provides an overview on the development of 3D navigation system for virtual outdoor and indoor environment using 3D game engine. Since the entertainment and game industry has invested so much in developing a powerful 3D engine to be used in various applications, this research work combined it with GIS components and optimized its functions to support the 3D navigation system in virtual environment. We also noticed that current 3D engines are not only provides rendering capabilities but also other functions such as video and sound support, improved AI with lots of skills (i.e. path finding and reactions/feedbacks).

We would like to address this 3D indoor and outdoor navigation in the near future by incorporating several enhancements to improve the 3D navigation system such as 3D buffering that allow users to generate relevant GIS analyses by creating 3D buffers in the virtual environment, highlights all possible destinations and display the information of buildings/rooms based on queries created by users and better users' friendly interfaces to ease users to use all of the functions provided by the navigation system.

REFERENCES

Coors, V. et. al (2004). Presenting route instructions on mobile devices. Book Chapter in "Geo-Visualization", 2004. Book Sections, 2004.

Coors, V. et. al (2006). A web 3D services for Navigation Applications. 1st International Workshop on 3D Geoinformation 2006. Kuala Lumpur, Malaysia.

Desney, T. et. al (2002). Exploring 3D navigation: Combining speed-coupled flying with orbiting. Microsoft Research.

Fritsch, D. (2003). 3D Building Visualization – Outdoor and Indoor application. Photogrammetric week. 2003.

Kada, M and Fritsch, D. (2004). Visualisation Using Game Engines. ,XX ISPRS Congress, Commission IV. Istanbul, Turkey.

Karas, I. et. al (2006). Automatically extracting 3D models and network analysis for indoors. 1st International Workshop on 3D Geoinformation 2006. Kuala Lumpur, Malaysia.

Meijers, M. et. al, (2004). 3D Geo-Information Indoors: Structuring for evacuation. ,XX ISPRS Congress, Commission IV. Istanbul, Turkey.

Stoter et. Al (2004). 3D GIS, Where are we standing?. ,XX ISPRS Congress, Commission IV, Istanbul, Turkey.

Tsai, Y.L. et. al (2000). An intelligent user interface with motion planning for 3D navigation. Proceedings of IEEE VR2000 international Conference.