COLLABORATIVE DESIGN: INTEGRATING GAME ENGINE TO ONLINE INTERACTIVE PRESENTATION TOOL

RAKTUM SALLKACHAT, ARAYA CHOUTGRAJANK
Master of Science Program in Computer-Aided Architectural Design
Rangsit University, Thailand.
nml@access.inet.co.th; araya@rangsit.rsu.ac.th

Abstract. In this research, we aim at the development of a computer-aided collaborative design tool, which supports the online architectural presentation independently from computer platforms and locations. Users are enabled to access the system and collaborate among working team via the Internet. Various online 3D Game engines are explored and evaluated. One 3D Game engine is selected and used as a base engine for the collaborative tool development. The application development will specifically concentrate on real-time presentation, shared representation and understanding among design teams and project owner. Traditional architectural design presentation in collaboration process will be imitated and transformed into digital age online presentation.

1. Introduction

Traditionally, design collaboration between designers and clients used to be simply managed by using sketch, two-dimensional drawing, architectural model, and perspective drawing. These traditional tools are essential in architectural design process since they represents some forms of prototypes, which help architects in understanding clients' needs as well as designing and developing projects that match clients' expectation. In this information age, computer technology has been playing an important role and gradually replacing the traditional tools. Outcomes from this new digital tool are so real that customers could convincingly experience the design prior to actual construction.

The development of architectural-specific application, especially 3-D presentation application, are complex and would need high-end system in both development and implementation phases. Customized software is usually expensive due to the requirement of large development team and time consumed. Various applications might be developed and integrated to support the complete presentation process.

Since the introduction of new media, so-called the Internet, online collaboration in design has been a major concern, especially in the area of online architectural presentation. Two problems are raised here.

First, architectural presentation is commonly displayed with still image or motion graphic in different file format, such as .avi, .mov. The parth of the animated scene is normally pre-defined. Viewers are forced to obtain information as provided by architects or designers. The capability of moving around freely in the virtual space and interact with the space are so limited that viewers would lose the sense of connectivity and adjacency of spaces.

Second, in order to get close-to-real experience, computer-generated photorealistic image must be created. The size of the image is typically enormous, results in bad quality image representation or unacceptable performance via the Internet.

2. 3D Applications and Quake

3D applications could be developed using different methods as follows:

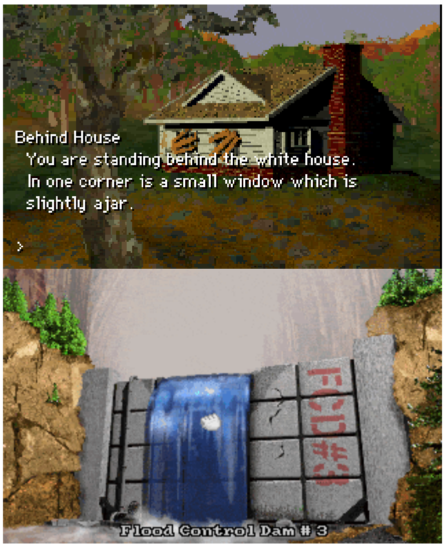
- Develop real-time rendering engine by implementing graphics software development kits (SDK), for instance, OpenGL and DirectX.
- Develop application based on commercial 3D graphic engine.
- Develop application based on 3D game engine, for instance, Quake II, Quake III Arena, and Unreal Tournament. These FPS games normally provide tools for scenes creator and game development kits (GDK).

Presently, computer 3D games (3D Shooting, 3D First Person Shooting) are able to generate realistic real-time interactive 3D spaces. Players could have full control over the movement in virtual scene in a multi-player environment (Shiratuddin and Thabet, 2000). General system environment could be simply implemented using TCP/IP protocol and inexpensive hardware. Many game developers allow users to further develop applications on top of their game engines. Users can improve or modify scenes, rules, and characters' behavior. This open source concept has brought developers and online game players to add their creativities to the games, hence convinced players to play with the games continuously. The example of these game engines are Doom and Quake (Quake, Quake II, Quake III) from id software, and Unreal Tournament which provides GDK (Game development kits) for modification.

Fundamental characteristics of computer online 3D games are similar to those of 3D scene generator applications used in architectural presentation.

They provide real-time realistic 3D generator, interactive information retrieval, simple hardware implementation, and multi-user capability via network. Only violent behaviors and weapon objects in 3D fighting games should be eliminated to enable players to interact with the spaces in a more friendly way and suitable for architectural presentation.

While 3D First Person Shooting (FPS) gives a realistic virtual space and environment, other types of game, such as Adventure and RPG (Role Playing Game), provide better information management in the game environment. Combination of both characteristics in single environment could enhance architectural presentation with aesthetic sense as well as rich information and collaboration.



Behind House
You are standing behind the white house.
In one corner is a small window which is
slightly ajar.
>

Figure 1. Adventure Game: Return to Zork, Activision, 1995.

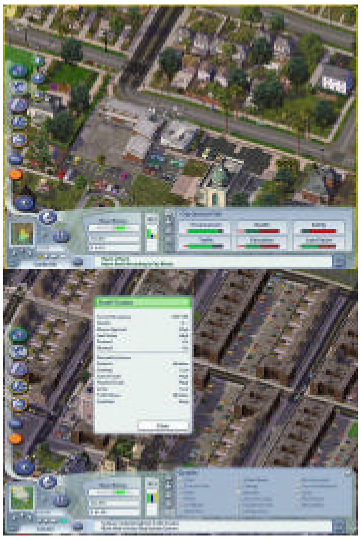


Figure 2. Simulation Game: SimCity 4, Electronic Arts, 2003.

Figure 3. First Person Shooting: Doom3, Activision, Id Software, 2003.

The research goal is to develop a prototype of architectural presentation application, which could be integrated into the online design collaborative environment. A 3D game engine, Quake III Arena, is selected as a base engine for the development. It provides Open Game Source Code that could be modified with C language, compiled to QVM (Quake Virtual Machine) and ported to different platforms. Functions and AI of players could also be modified to suit the characteristics and behavior of application.

Key concentration of this prototype development is the capability to perform the real-time interactive online architectural presentation via the Internet or TCP/IP network. Functions are designed to support the collaboration in architectural presentation process. User interface is studied and designed to best suit the application. Furthermore, specific data is integrated to objects in the scene in order to provide richer information and better experience to viewers. The developed application could be implemented on personal computer with affordable budget.

3. Quake III Engine

Quake III Arena is 3D First Person Shooting (FPS) from Id Software. Player could view a first person perspective by moving in 3D virtual world, and connect with other players via network or the Internet. Game environment, level/map, could be modified. Quake III Arena is a fighting game with violent behaviors and scenes. Open game source code allows programmer to modify behaviors, scenes, and rules (mods). The modification excludes engine source (i.e. rendering, network, etc.). Quake engine source code has been open to public under GPL (the General Public License) since 1999.

Quake III Arena Engine supports functions, such as shader curved surfaces, 32-bit color, special effects, networking and super-smooth speedy hardware rendering. With these functions, it is flexible to apply this engine in the development of various applications. CAVE Quake III is an example of application that applied Quake III Arena to be used on CAVE. Paul Rajlich had mentioned the advantage of Q3A as follow,

> "… In many ways, Quake III Arena represents the state of the art in real-time rendering. The CQ3A engine implements many of the features that are in the real game including multi-pass shaders, curved surfaces (bezier patches), bsp tree with pvs testing/culling, lightmaps, animations, skybox rendering, etc
>
> Even if you don't care about gaming, the Quake III format is a great format for creating original 3D content. There are several free map editors available, including the incredibly

> powerful QERadiant. Now you can easily import worlds created with these tools into the CAVE! Check out my house model!
>
> It is open-source (GPL) and cross-platform. I restructured the engine so that it can handle multiple rendering contexts and multiple rendering pipes. For details, read this short VR source article. The result is that CQ3A can compile with 7 different interfaces! Each interface is small and completely separate from the rest of the engine. The interfaces are listed on the Downloads Page." (Rajlich, 2000)

Figure 4. CAVE Quake III Arena, Paul Rajlich, http://www.visbox.com/cq3a/

4. Quake III Arena Technology

Q3A working mechanism is the combination of Q3 Engine and Game Engine. The game engine is consisted of 3 modules, which are cgame module, game module, and q3_ui module. While the cgame module controls screen output on client-side, the game module controls functions of game server, connections among players, and bot functions Bot is an abbreviate for robot, an artificial player for the computer game (Waveren, 2001). The q3_ui module controls all menu functions.

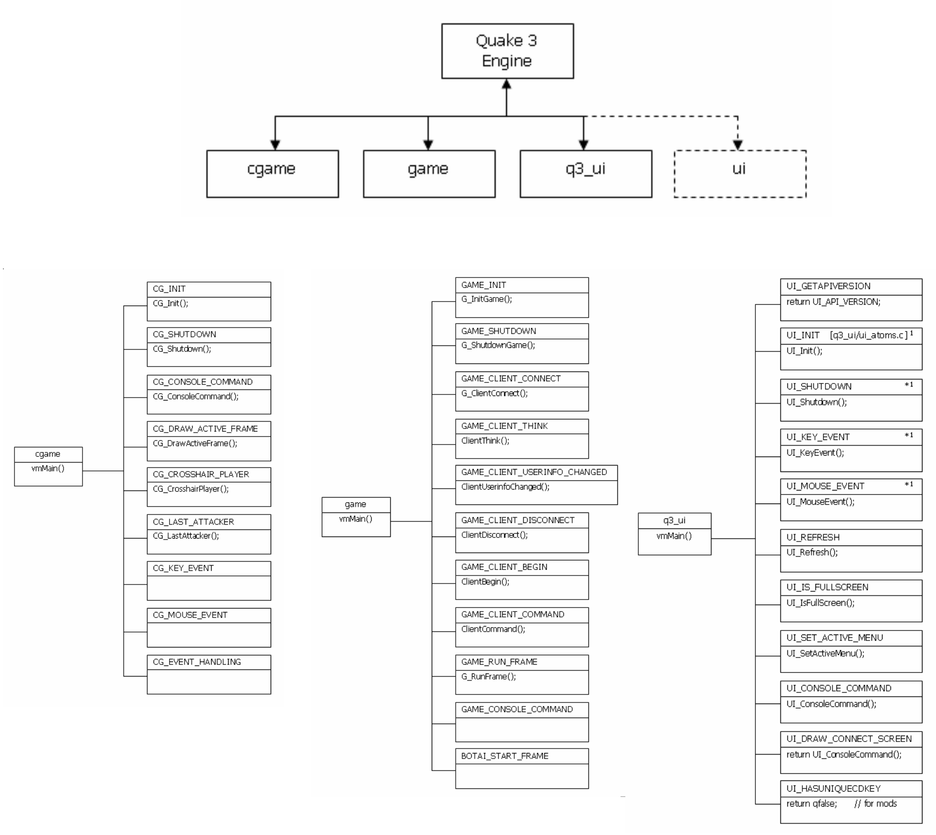


Figure 5. Q3 Game Engine Diagram.

For scene construction, Q3A preprocesses the scene environment, main lighting, and bot movement for better performance in real-time rendering, especially in large scene or large architectural scene development. (Dinesh, 2000). Level of Detail (LOD) calculation is executed to reduce the polygon numbers when player object is moving far away in the scene. This calculation assists in accelerating multi-player interactive.

The development procedures of application based on Q3A engine include:

- Modify program source code to suit architectural presentation functions and compile it to QVM (Quake Virtual Machine) format,
- Create architectural scene, building, and decorative items with texture, lighting, and finishing materials mapping following these steps:
  - Create building model with GtkRadiant,
  - Create decorative items in low polygon,
  - Prepare material texture and build shader for special texture.
- Create player model (bot) with appropriate behavior.

5. Program Modification

In the development process, *The CAAD Lab* (Computer-Aided Architectural Design Lab of Rangsit University) was chosen as a model for research experiment. Virtual 3D scenes, finishing material textures, and special effects were created. Decorative items were also created and stored in object library. Functions and behavior of objects were modified to fit the architectural process.

5.1 SCENE CREATION

2D drawing and interior design were transformed into 3D model with GtkRadiant. In this experiment, we specified "32 Quake units" an equivalent to "1 meter".

5.2 MODEL CREATION

All decorative objects such as table, chair, and lamp were produced in low polygon, imported/exported with 3D Studio max and Plugin. Index keys were generated with GtkRadian, and stored as objects were placed in the virtual space. Objects information was pointed with these index keys and was displayed on the screen when activated.

5.3 MATERIAL/TEXTURE CREATION

Finishing material textures such as floor, and wall were created with special effects. Reflecting texture, moving texture, lighting objects were built with Q3A Shader.

5.4 BOT/AVATAR CREATION

Bot execution in Q3A was studied for the basis of future development. In this research, a cartoon character was created to study movement and behavior. Human character could be researched and developed further to get more appropriate behavior for each collaborate classification.

6. Architectural Presentation Process

The development of application based on Q3A game engine to support architectural presentation process must focus on the way architects present their works. Objects, materials, decorative items, and texture must be prepared and stored in a library. Stored objects could be easily used and reused. Started with 2-D drawing, map editor software (i.e. GtkRadiant, gmax) is used to identify objects, materials, and lighting. Identified object is compiled to Q3A scene (.bsp), and then, copied to Q3A server. At this point, players or users could access the server, download the scene and interact with the pre-set scene in real-time. The system will detect client machines and will automatically download the most updated scene while users are moving around the virtual scene freely. Collaborative functions are also available to enable users to work online as a team. These functions include chat, inquiry, key map, users' location indicator, and room information.

For more appropriate practice in professional collaboration, game source must be modified. Violent behavior, security log in, key map, movement mode, and collaborative interface, are examples of functions added or modified in this research prototype.

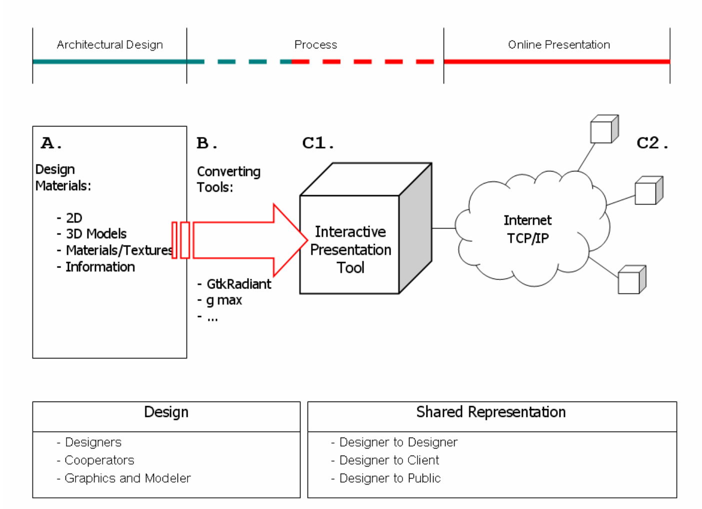


Figure 6. The overall process to develop application from 2D floor plan.

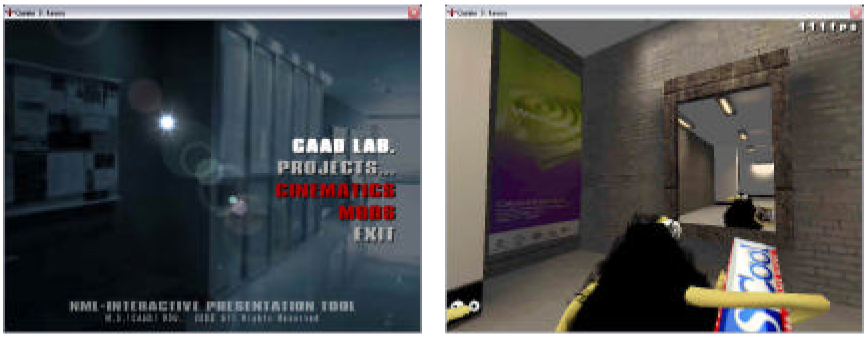


Figure 7. Interactive Presentation Tool Using Q3A Engine

6.1 VIOLENT BEHAVIOR CLEAN UP

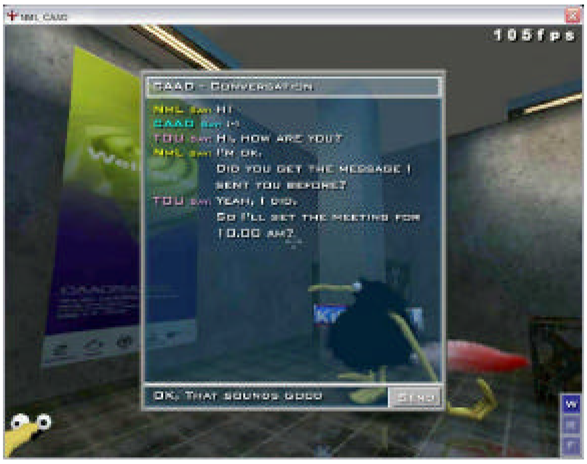
The unwanted violent behavior of original game could be removed by adjusting code in *cgame* module. New scene could be created without any weapon attached or new player model could be created without violent behavior.

6.2 USER INTERFACE DESIGN

Collaborative application should provide security check and tracking capability to ensure access right privilege control and information log record respectively. Login function is implemented and information is tracked through users' activitie s. User interface for chat, movement mode, key map, and location indicator are designed and developed.

6.3 ADJUST MOVEMENT MODE

Movement control in virtual space is displayed in 3 different modes: walk, run and fly. Lower part of the screen displays character name, movement mode, and instructions.



CAAD - CONVERSATION

NHL HI

CAAD HI

YOU HI, HOW ARE YOU?

NHL I'M OK.
DID YOU GET THE MESSAGE I
SENT YOU BEFORE?

YOU YEAH, I DID.
SO I'LL SET THE MEETING FOR
10.00 AM?

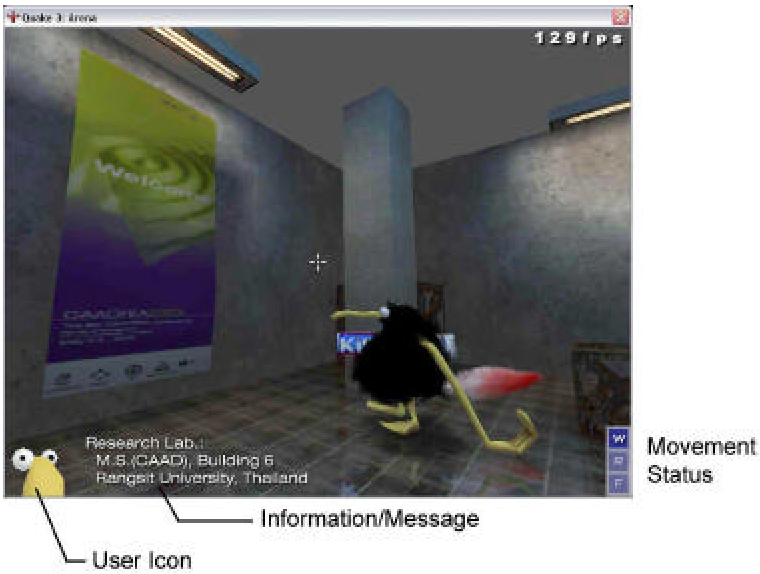
OK, THAT SOUNDS GOOD

Figure 8. Heads Up Display (HUD) and Chat Screen.

6.4 INFORMATION RETRIEVA L

To avoid the lost of direction, key map on screen is introduced to assist users in identifying current location and other users' location while moving in the virtual space. Users can walkthrough the space using this key map as a direction guide. Key map is generated and displayed on screen constantly with the coordinate intersection computation technique.

Design Collaborative via the Internet could be classified in two modes:



Research Lab.:
M.S.(CAAD), Building 6
Rangsit University, Thailand

Figure 9. Heads Up Display (HUD).

synchronous and asynchronous (Maher, Simoff and Cicognani, 1999). In general, most game engine interaction is in synchronous mode. Compared with architectural presentation, walkthrough and chat in virtual architectural space would be categorized in the synchronous communication. For asynchronous mode, more functions could be provided, such as environment information, objects detailed information, message board, or navigator bot that provides instant messages to viewers.

7. Further Development

From this research, advanced functions could be designed and developed to serve as better collaborative tools for architects, educator and related professionals. Some suggestions for future development are as follow:

- add more functions to cover entire architectural design collaboration,
- develop functions in both synchronous and asynchronous modes. Functions include bulletin board system, instant messaging, or comment notation,
- develop application to support different languages, especially Thai language. At present, Q3A font is encoded in 7 bits, which does not allow Thai Language to be used. We can use the same technique as UUEncode which transform 8 bits characters into 7 bits encoding before broadcasting,
- develop tools for importing CAD file from CAD software (i.e. AutoCAD) into the proposed application,
- Use Quake II engine, the previous version of Quake, as a base development engine if main engine modification is needed. Quake II's entire source code is available for more complete development. Rendering system, network system, and special input mechanism could then be modified. Alternatively, source of other games that are developed on Quake III engine could also be used to improve quality of the application,
- Implement public internet server to serve public projects.

8. Conclusion

The application model and prototype developed in this research supports conventional architectural presentation process. Benefits of online presentation are relying on the possibility of users to both communicate with others and interact with virtual architectural space. Multi-user capability allows collaborative design among team members in different locations and cross platforms, which results in significant reduction of communication cost.

Choosing available online 3D game as base engine for application development could also cut down the development time; hence user-specific application could be developed with minimum budget. Furthermore, online 3D game interface and environment are already well-known to wide range internet viewers. Users could be able to get familiar with the application in no time.

This research concludes that the application development based on 3D game engine could be a practical approach in the development of other support tools for architectural design process. Besides the prototype of collaborative tool, we have an opportunity to study 3D graphic engine in depth. It is convincing that the rapid growth of online community and the popularity of online 3D game could drive the continuous development of various open source CAAD applications in the near future.

References

Campbell, D. A. and Wells, M.: 1994, A Critique of Virtual Reality in the Architectural Design Process, University of Washington, Technical Report: R-94-3, http://www.hitl.washington.edu/publications/r-94-3/.

Churchill, E. F., Snowdon, D. N. and Munro, A. J.: 2001, Collaborative Virtual Environments, Digital Places and Spaces for Interaction, Springer, Athenaeum Press.

D!ABLO: 2002, D!ABLO's Map-Editor, http://www.shadowflare.com/.

Dodge, M. and Kitchin, R.: 2001, Mapping Cyberspace, Routledge.

Joseph, W.: 2001, Q3map explanation, http://www.planetquake.com/spog/stuff/technical.html.

Maher, M. L., Simoff, S. J. and Cicognani, A.: 1999, Understanding Virtual Design Studios, Springer-Verlag London Berlin Heidelberg.

Manocha, D.: 2000, Real-time rendering (A Framework for the Real-Time Walkthrough of Massive Models, Department of Computer Science, University of N. Carolina, Chapel Hill, http://www.cs.unc.edu/~hoff/papers/mmr/i3d99.pdf.

Phaethon: 2002, Baseq3 source commentary, http://www.linux.ucla.edu/~phaethon/q3mc/q3mc.html.

Planet Quake: Dedicated to Quake III Arena, http://www.planetquake.com/quake3/.

Proudfoot, K.: 2000, Unofficial Quake 3 Map Specifications, http://graphics.stanford.edu/~kekoa/q3/.

Radford, A.: 2001, Let the games begin: philosophy, pedagogy and pragmatism, Playful Learning Forum, Adelaide, Australia, http://www.arch.adelaide.edu.au/games/forum/papers/KN_LetTheGamesBegin.pdf.

Rajlich, P.: 2000, Cave Quake]I[Arena, http://brighton.ncsa.uiuc.edu/~prajlich/.

Rouse, R. III: 2000, Designing Design Tools, Gamasutra March 29, 2000, http://www.gamasutra.com/features/20000323/rouse_01.htm.

Shiratuddin, M. F. and Thabet, W.: 2000, Virtual Office Walkthrough Using a 3D Game Engine, Department of Building Construction, VirginiaTech, USA, www.arch.usyd.edu.au/kcdc/journal/vol4/shira/ijdc_final_journal.pdf.

Waveren, J. P.: 2001, The Quake III Arena Bot, Delft University of Technology, http://www.kbs.twi.tudelft.nl/Publications/MSc/2001-VanWaveren-MSc.html.

Yu-Shen Ng: 1997, Internet Game Design, Gamasutra, August 1, 1997, http://www.gamasutra.com/features/19970801/ng.htm.