Design of a Multiuser Virtual Trade Fair Using a Game Engine

I. Remolar, M. Chover, R. Quirós, J. Gumbau, P. Castelló, C. Rebollo, and F. Ramos

Institute of New Image Technologies,
Universitat Jaume I,
Castellón, Spain
{remolar,chover,quiros,jgumbau,castellp,rebollo,jromero}@uji.es

Abstract. The current world economic situation makes it necessary to develop new ways of establishing commercial relationships. One possible solution is to explore the advantages of virtual worlds, and for this reason online virtual trade fairs are becoming more popular in the business world. They enable companies to establish a trade relationship with their customers without the need to visit them in person. This is very attractive for exhibitors because it can save them money, which is a priority for many companies today. In this line, this article presents a multiuser virtual trade fair developed using 3D game engine technologys. Users represented by avatars can interact with each other while they are visiting the virtual fair, which has some interactive objects included in the stands to provide information about the exhibitors. This virtual world is accessible online, and visitors only require a plug-in on their computers to be able to enter the virtual world. The game technology makes it possible to obtain a high degree of realism: very real lighting, cast shadows, collision detection, etc. Moreover, the virtual world presented builds the 3D objects automatically. Participants in the trade fair can customize their virtual stand and the application will generate the code necessary for its inclusion in the rendered virtual world.

Keywords: virtual worlds, game engines, web application, business application, multiuser environment

1 Introduction

Virtual business is currently an interesting way to increase sales and income. This is the main reason why virtual trade fairs are becoming so popular: *"make real money in a virtual world"* [1] [2]. In these digital events, customers and sellers can meet in a virtual scenario. The companies show their products and provide information about the services they offer [3] [4] [5]. Virtual trade fairs have actually many benefits more than regular trade shows [6]. They are more accessible than a physical exhibition, as they are available 24 hours a day. A huge benefit is the reduced cost of participating in a virtual trade fair as opposed

to a physical one. There are no travelling costs, nor any costs associated with employing staff for maintaining the trade show area.

The "Virtual Tradeshow Survey" [7] recently published, demonstrates how important is it for show organizers to have virtual events as part of their portfolio in order to truly engage an attendee in a year round environment. It also offers a comparison between physical events and virtual events from the collective standpoint of an event organizer, exhibitor and attendee and it explains how virtual events enhance other event attendance.

For all these reasons, the number of virtual trade fairs available on the web is increasing nowadays. Regarding the technology they use, most of them are not based on 3D models [8] [9]. Only very few virtual worlds are composed of 3D objects. These kinds of objects increase the realism considerably. The most popular technology used in these worlds is VRML or X3D [10]. Another widely used technology is the 3D engine of Adobe Shockwave. The virtual fair we present in this article has been implemented using the technology offered by game engines [11] [12] [13]. This type of software provides a suite of visual development tools that can produce very realistic scenes. Moreover, they allow us to design multiuser environments where some phisical effects are easily included. Moreover, game engines provide the necessary code to include in the virtual scenes players, that programmers can used to represent the users. Figure 1 shows a screenshot of the virtual fair. Up until now, game engines have been mainly used in the game industry. However, these tools make it easy to create virtual worlds, whose purpose is more serious than entertainment alone.

Our virtual fair has some characteristics that make it interesting to the business world. The most important one is the easy method implemented to manage this fair. Someone with no computer knowledge can build a scenario in it because the 3D objects appearing in the fair are automatically built [14] [15] . The participating companies can configure a virtual stand, adding pictures, contact information, and so on, and the application will generate the code necessary to create it, taking all this information into account. The main objective of the virtual fair is to provide publicity for the companies that have a stand within it. This has been taken into account and some selectable objects have been included in the stand of the trade fair. These objects can be configured by the companies to enable information about their business and the products they offer to be accessed by just clicking on these selectable objects.

The graphics hardware available in companies was a very significant obstacle. Not all businesses had the latest graphics cards installed in their computers. So this fact restricted the programming of our application. The latest technology in graphics programming could not be used because it would have prevented most companies from rendering the virtual fair. The virtual fair we are presenting can be displayed on the type of computers that most companies have.

In order to make a visit to the fair friendlier, users are represented by means of avatars, i.e., virtual characters. Users can customize these features: the gender, the clothing, etc. Moreover, various users can be connected at the same time and they can communicate with each other [16] [17]. All of them show their username

Fig. 1. The outdoor environment of the virtual fair.

in order to be identified. Text-based messaging has been implemented in order to make this communication possible, as in [18].

The virtual fair is very easy to visit, because it is accessible using an Internet browser [19]. End users do not have to install specific software in order to connect to it. When they first visit the website, their browser will automatically download and install a plug-in, making the connection process transparent to the user.

In the next section related work is reviewed, then the system architecture of the virtual fair is explained in Section 3. Section 4 deals with the virtual trade fair management, and the automatic creation of the trade fair is detailed in Section 5. Sections 6 detail the web environment and the interaction in this virtual world. The client-server organization is explained in Section 7. Finally, some conclusions and future work on this application are presented in Section 7.

2 Related work

The commercial potential of the new virtual worlds is impressive and makes them well worth a look. Currently, many new virtual trade fairs are appearing, even companys that offer to develop customized virtual trade fairs, such as BusinessGlobal [20]. Nowadays, most companies are looking for cost-cutting, so they are seeking new sales opportunities and new ways to show their products and services to possible customers using a global communication channel such as Internet. The main requirement of these virtual worlds is that they have to be remotely accessible by typical computer hardware configurations of companies not dedicated to computer graphics.

This requirement determines the technology used in building virtual worlds. Many of the virtual fairs that have appeared up until now are modeled in 2D technologies [8] [9]. However, this produces a lack of realism in the virtual world. Other projects model 3D objects [21] [20], but the best realism is obtained using a

game engine [10]. For years now, the huge and growing computer games industry has helped drive the innovation of certain technologies, particularly graphics and user interfaces. Therefore the use of this technology to design virtual fairs makes these virtual markets more attractive to users.

Moreover, online gaming allows people to interact together in a computer game over a network. Huge virtual worlds are continually running on hundreds of servers for users to interact with each other and the virtual world they inhabit. This aids in building a virtual world where people can interact.

The most popular market in a 3D world is Second Life [1]. This is the most visited 3D virtual world inhabited by millions of residents from around the globe. As it is a community, albeit a virtual one, there are as many opportunities for innovation and profit in the Second Life world as there are in the real world. Thousands of residents are making part or all of their real life income from their Second Life businesses [2]. However, Secondlife is a very general and open virtual world. Real business are looking for some personalized virtual environment where the access is controlled by an administrator and has been designed according to its requirements, the same as a regular trade show. Summarizing, our objective with this virtual fair is to promote real business in a virtual world.
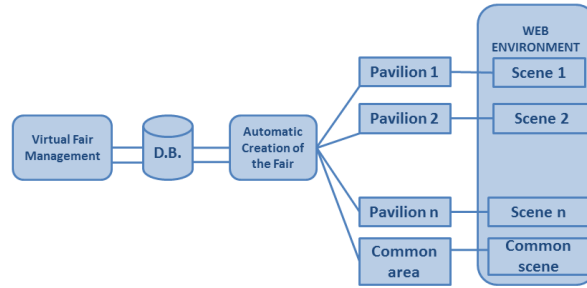
Fig. 2. The organization of the different modules that make up the developed application.

3   System Architecture

The main components have been organized into three different modules that have different tasks, as can be seen in Figure 2:

– *Virtual Fair Management*: this allows the fair administrator to manage the access of the companies to the virtual world. All the information about the exhibitors is stored in a database to allow the configuration of the virtual scenarios. In addition, this module enables the participant companies to configure their stands in the fair. They can easily register, choose the type

of stand they prefer, add some multimedia content and choose the location and orientation of the stand in a pavilion.

– *Automatic Creation of the Fair*: this module automatically creates the pavilions from the information previously entered by the companies. Pavilions are the buildings where the virtual stands are located. They are configured in independent virtual worlds, which are usually called missions in game engine terminology.

– *Web environment*: this module provides Internet access. It enables the selection of avatars and manages the rendering of the exhibition and the clients through the Internet server. It is the module that makes it possible to access the virtual trade fair from a computer with an Internet connection.

– *Client-Server module*: this allows to implement a client-server model required by the game engine used in this application: Torque Game Engine (TGE) 1.5.2.

4 Virtual Fair Management

This part of the application makes it possible for users without specialized computer training to manage the 3D fair. Two kinds of users can access it: the fair administrator and the business administrator.

4.1 Fair administrator user

As shown in Figure 3, after loggin in, the fair administrator can choose from a series of actions for efficient fair management. In brief, these actions include management of companies, assignment of areas in different pavilions for the companies and stand configurations. Figure 4 shows an example of the interface used for creating a company. It is easy to observe what type of fields are required for a company to be successfully registered.

Fig. 3. Available options for fair administrators.

Thus, the main role of the fair administrator is to enter general information about the participating businesses into the database and to provide them with a password so they can access the application. The other task of this user is to manage the location of the stands in the pavilions. A layout of available spaces in these pavilions has been designed as a square matrix. Each space has a virtual size of 16 m^2. This layout accommodates the three types of stands designed.

The smallest one includes one square, i. e. 16 m^2, the medium one includes two squares, i.e. 32 m^2 and the largest one includes four squares, i.e. 64 m^2. The application shows the occupied stands colored red and the empty ones colored green. Figure 5 shows an example of available and occupied stands within a pavilion. It is important to note that the label "Access" shows where visitors will enter and leave the pavilion.


Fig. 4. Creating a company in the Virtual Fair Manager.

In order to position the stand for a business, the fair administrator simply selects as many empty squares as are needed for the size of the stand, and then checks its orientation.

4.2   Business administrator user

After the information about the participant is stored in the database, the fair administrator issues a username and password to the business administrator. The business administrator is responsible for configuring the final appearance of the stand and customizing the content displayed in the virtual environment. Once the size of the stand has been chosen, there are three different stand models available for selection. In this step, this user perform two important actions. In the first stage, companies select a color palette and a design from some predefined models (Figure 6). Later, at the second stage, companies can freely customize their stands by combining different type of multimedia items (Figure 7).

Figure 8 shows an example configuration of each one of the different sizes, 64, 32 and 16 m^2 stands. The customization also enables different color combinations to be selected for the stand. Moreover, the business administrator can include some multimedia information for display in the fair. This is stored in the database
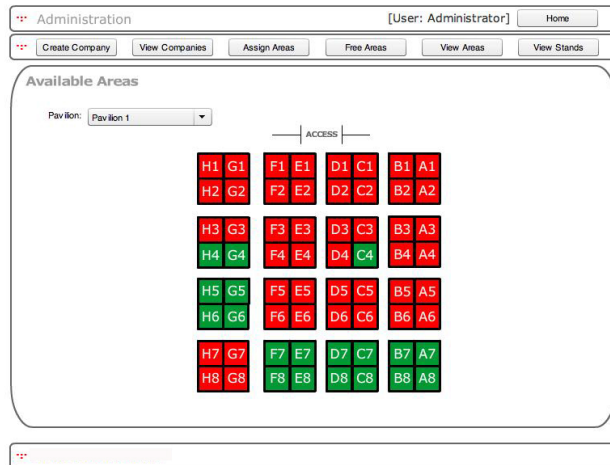
Fig. 5. Available and occupied areas within a pavilion.
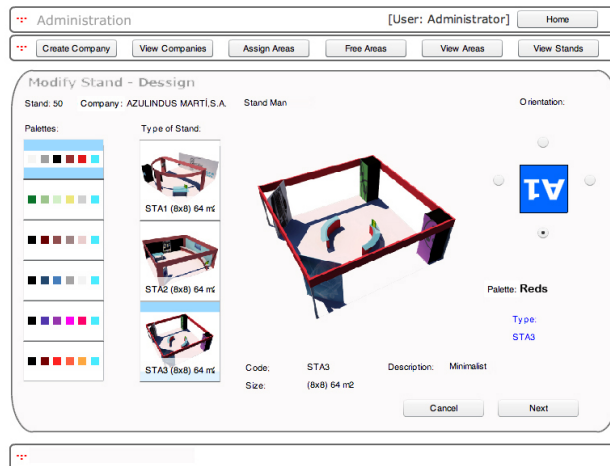
Fig. 6. User interface for basically configuring the stand of the company.
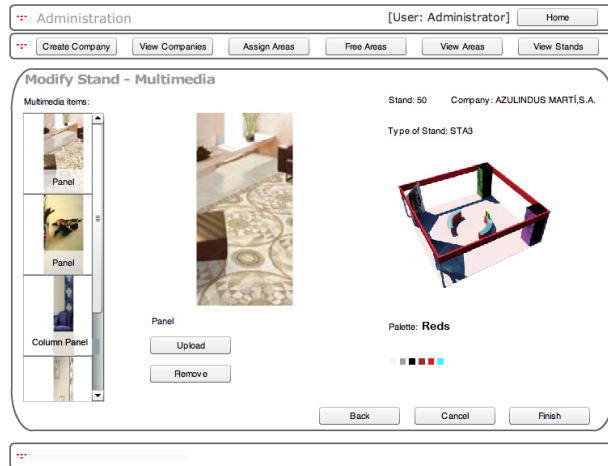
Fig. 7. User interfaz for updating some images or textures for the stand.

and it may include pictures of products and/or services multimedia materials and Internet information, such as the website address of the business.

To implement this application, PHP and MySQL were employed. This framework has a series of advantages such as low cost, ease of use, HTML embedding, cross-platform compatibility, stability, speed, open source licensing, many extensions, fast feature development and strong user communities for supporting.

A MySQL database is used to store all the persistent information, PHP is used for efficient interaction between the database and client applications and a Flash movie embedded into a web page is used for administration. The combination of these technologies makes it possible to automatically manage the virtual world displayed in the game engine through a web environment. Both types of users are able to easily access the administration application, as this only requires a web browser with a Flash player.

5   Automatic Creation of the Trade Fair

The game engine employed in this project, Torque, has conditioned the organization of the information. Torque implements every game following a client-server model, even if there is a single player. This concept will be extended in Section 8. This client-server architecture conditions that the files of the project are organized in three main categories: files for the *server* module, files for the *client* module and the general *data* of the project. The first time a user connects to this virtual world, the folders *client* and *data* are downloaded to his/her computer. The final organization of the files involved in this virtual world is shown in Figure 9.
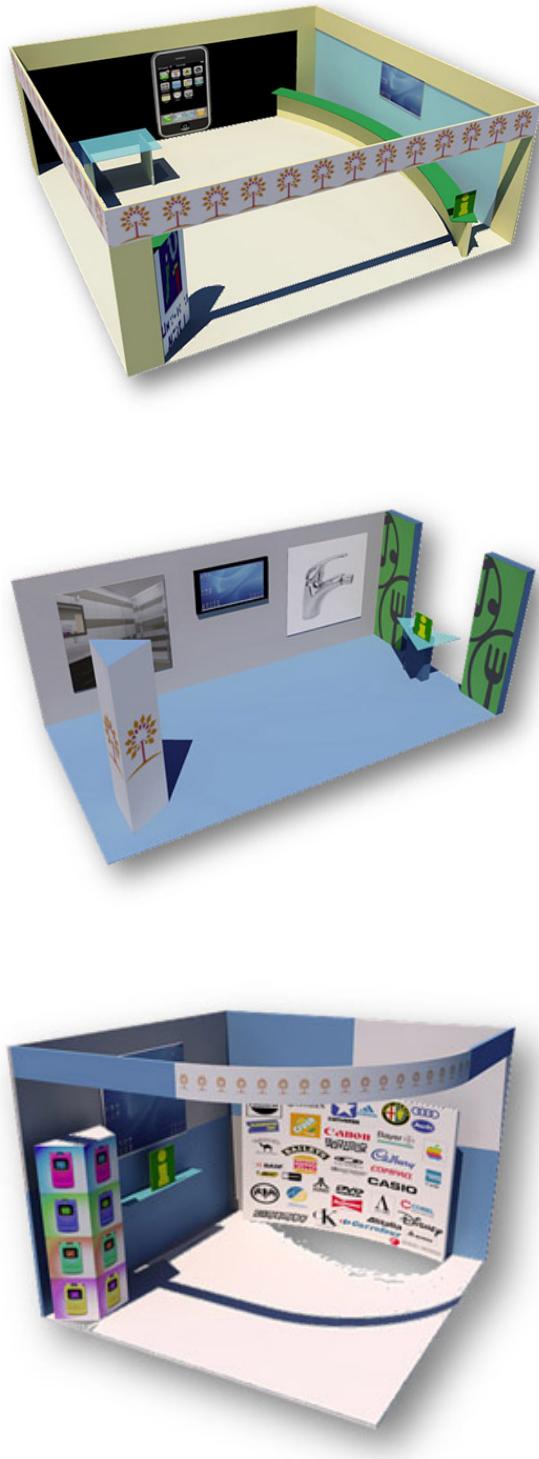
Fig. 8. Different virtual stand configurations, 64 m^2 on the left, 32 m^2 in the middle and 16 m^2 on the right.
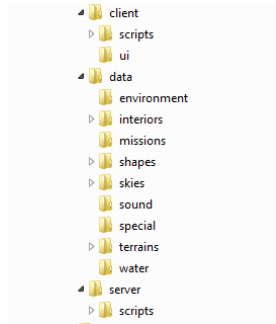
Fig. 9. Files organization in the virtual trade fair.

In the virtual trade fair, two main classes of 3D objects, following the requirements of Torque, have been designed. The first type is composed by the 3D objects where the player can move around, as buildings or outdoor environments. The other type is integrated by the rest of the 3D objects, objects that the designer can add to the virtual world in order to make it more attractive. In the first group, this virtual trade fair has two objects, the outside environment and the pavilion. They both have been modeled with a specific tool such as *Torque Constructor* [23]. These type of objects are called *interiors* in the Torque terminology. The other kind of objects have been modeled with the 3D Studio Max 2009 [24]. They all are stored in the *shapes* folder.

As it is mentioned before, three different stands have been created for every size. All the geometry and the necessary files for their configuration are stored in the folder */shapes/stands*. Let A be the smallest stand, B the medium and one C the largest one. Three different versions of every size have been modeled. Every folder in */shapes/stands* stores all the data involved in the render of the type of stand specified by the letter and number that appear in the name of every folder. The organization is shown in Figure 10.

The scenarios are built from the information stored in the folder *missions*. Initially, one file is stored in this folder for every basic scenario. These files, also called missions in the game engine, have some typical components of an outdoor scene, such as the sky, the sun, the sea and the terrain, and the position where the player is going to appear in them (*SimGroup*). The basic mission of the outdoor environment has the six pavilions. Nevertheless, the initial mission of the pavilion is the same for all the pavilion scenarios, and later it is configured according to the one is going to be rendered. Briefly, this code is shown next:

```
new InteriorInstance(Pavilion) {
  canSaveDynamicFields = "1";
  Enabled = "1";
  position = "0 0 0";
  rotation = "1 0 0 0";
  scale = "1 1 1";
```
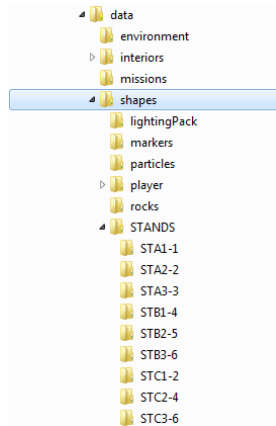
Fig. 10. Files organization in the virtual trade fair.

```
    interiorFile = "/data/interiors/pavilion/PB0001.dif";
    showTerrainInside = "0";
    smoothLighting = "0";
};
new Sky(SkyObject) {
    ...
}
new Sun(SunObject) {
    ...
}
new TerrainBlock(Terrain) {
    ...
}
new SimGroup(PlayerDropPoints) {
    ...
}
new WaterBlock(WaterObject) {
    ...
}
```

The module in charge of the Automatic Creation of the virtual fair processes the database and automatically generates a mission for the scenario the avatar is going to appear. If he/she is going to be in a pavilion, the mission that generates this scenario is configured with all the stands and geometry that the fair and the businesses administrators have included in it. The generated code is added to the initial mission file in order to render the final scenario with all the changes that the users have included. This code establishes the type of stand, its position, its rotation and some details about the lighting of this stand. One example of

code that renders this initial stand is briefly detailed next. The name assigned to it determines the pavilion and its position in it. In the following example, the stand is placed in the pavilion 1 (P1) and in the area labelled by F1.

```
new TSStatic(st-P1-F1) {
    canSaveDynamicFields = "1";
    Enabled = "1";
    position = "226.252 -0.353153 16.6096";
    rotation = "1 0 0 0";
    scale = "1 1 1";
    shapeName = "/data/shapes/STANDS/STA1-1/sta1-1.dts";
    receiveSunLight = "1";
    receiveLMLighting = "1";
    useCustomAmbientLighting = "0";
    customAmbientLighting = "0 0 0 1";
    usePolysoup = "1";
    allowPlayerStep = "1";
};
```

The next step is to customize the stand following the requirements established by the Business administrator. This is possible by adding some code to the current mission. One example of code is shown next: the stand named *st-P1-F1* will change the default texture *P01M02.png* by the established by the business user, *NewTex*.

```
nametoid("stA1").ResetDynamicSkin("st-P1-F1","P01M02.png","NewTex");
nametoid("stA1").UpdateDynamicSkins();
```

Every time a user connection is established, this module accesses the database to create the stands configured in a previous process and the code is generated to render the selected type of stand in the position and orientation configured for each business by the fair administrator. Moreover, multimedia objects in the stand are assigned a link to the business web page where the end user can obtain more information about their products and services.

Once this process ends, the necessary structure of files is created and it is possible to render the scene where the configured stand appears inside the pavilion. If something has been changed from the previous connection, the different textures and materials as configured by the business administrator are copied to the game engine file structure in the final user's computer, and assigned to the stand generated.

This part of the application has been implemented in PHP using the MySQL database.

6   Web Environment

All the modules described above are managed over the web, so the main function of the Web environment is to provide a website where the virtual world can be

accessed by the virtual fair users. This module has been implemented using different web technologies such as HTML, JavaScript and PHP with a MySQL database and a web browser plug-in.

This module performs various tasks. One of them is to control the access of visitors to the 3D world. The module checks the data stored in the database in order to allow entrance to the virtual fair. All the users have to be registered in advance. If the user is a visitor, before connecting for the first time, s/he has to fill out a short questionnaire. Figure 11 depicts a screenshot of the user registration form. The application stores this information in a MySQL database and then allows the user to enter.

Fig. 11. A screenshot of the user registration form.

Each user in the virtual world is represented by an avatar. There are four different avatars available in the web environment: two male and two female, also shown in Figure 11. Moreover, users can choose the type and color of their avatar's clothing. This process allows users to complete their profile information, which can be changed later if necessary.

The final integration of the virtual world on the web is done through the client-server module. It is responsible for the communication between the client and the server of the game engine. As mentioned above, the game engine chosen was the Torque game engine, TGE, [22]. It is a modified version of a 3D computer game engine originally developed by Dynamix for the 2001 Tribes 2 FPS game. The Torque engine provides robust networking code, scripting, in-engine world editing and GUI creation. Moreover, the source code can be compiled on

most popular platforms, including Windows, Macintosh, Linux, Wii, Xbox 360 and iPhone. This game engine also allows environment mapping, Gouraud shading, volumetric fog, and other effects such as decals that enable textures to be projected onto interiors in real time.

The version of the Torque game engine used in the implementation of the virtual fair is TGE 1.5.2. This version does not support web browser integration, so an external tool (plug-in) capable of integrating a game application into web browsers has been used. This tool is called *Igloader* [25]. *Igloader* is a cross-browser plug-in that allows virtually any Windows- based software to be delivered and operated within the browser window.

A light version of the Torque game engine (TGE 1.5.2) was used in order to lower the hardware requirements needed to run our virtual fair. We had to adapt the game engine code to make it compatible with the plug-in. The latest version of the game engine Torque (Torque 3D) can be integrated into some web browsers without any external plug-ins. However, this latest version is very complex, so the minimum hardware requirements are very high. This means that the graphics card installed on the user's computer system must support all recent programmable graphics processing units (GPUs) features such as shaders, which significantly limits the number of potential users in a web environment. This could be problematic for our application because even users with basic computer systems must be able to use the virtual trade fair over the Internet.

The *Igloader* plug-in can control the behavior of the application that runs in the browser by passing in parameters specified in an XML configuration file which can be created before the application starts running. In this way the web environment module is able to pass the type of avatar, its name, its position in the virtual world, etc. to the client game engine. The plug-in also has a powerful data streamer at its core that can be utilized to stream content in the background and patch current installations.

The web environment is hosted on a separate server to the one hosting the game engine. This prevents the server game engine from being affected by the load generated by the web environment. Each instance of the game engine can only run one mission, in our case pavilion or outdoor environment, thus several instances are running with a different TCP/UDP port and waiting for clients to connect to the virtual environment.

6.1   Interaction in the virtual fair

Different types of interactions have been implemented in the virtual fair. By default avatars can access the common area in the virtual fair. They are labeled with their username, as can be seen in Figure 12. The user can move around the virtual world with the avatar, using some on the keyboard to walk around and visit places of interest.

From the common area, the avatars can access the pavilions through the gateways (Figure 12). These gateways allow the user to leave the scenario s/he is visiting and to automatically appear in another one. The application quits the client's current mission of the client and downloads the new one. In the

XV
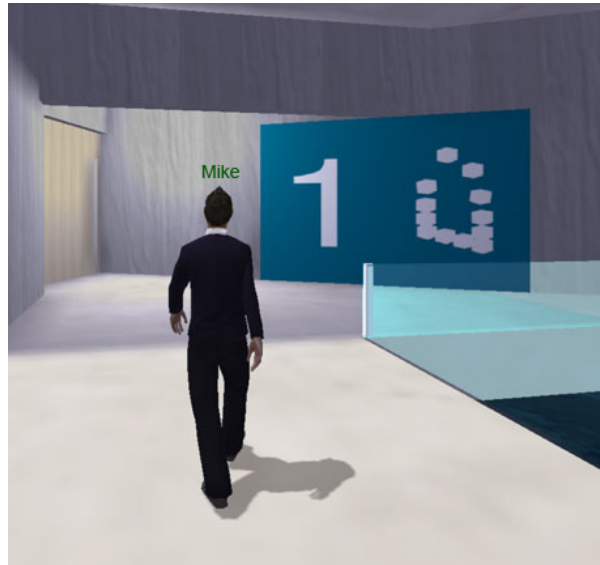
Fig. 12. A screenshot of the application, showing where the gateways are located in the virtual world. In this screenshot, a user is approaching the entrance to a pavilion.

virtual fair, these gateways are usually placed at the entrance to the pavilions. Another way of getting to a stand quickly is through the trade fair directory. The position of the stand and its direction are retrieved from the database, and then the avatar is placed near the selected stand and looking towards it. This information is passed from the web environment module to the client-server module of the 3D application, which is responsible for uploading the mission selected and placing the user in the correct position.

More interaction with the 3D world is possible through special objects located in the stands. These allow the user to view information about the company as well as its website address, e-mail, etc.

One of the main features of this 3D virtual business fair is the text-based communication it offers through chat. The chat system is part of the client-server architecture of the Torque game engine. It takes advantage of the multiuser connections to the virtual world and it allows the user to send and receive text messages. Figure 13 shows an example of chat. To send a message, the user only has to click in the dark area in the lower left corner of the window and type. The other users connected to the virtual fair will be able to read the message when the writer presses the enter key. Just above this area there is a display of the messages sent by users.

Fig. 13. A screenshot of the interior of a pavilion, showing text-based communication between users.

7   Client-Server Module

The 3D game engine used supports networked games over a LAN and the Internet with traditional client-server architecture. Server objects are "ghosted" on clients and updated periodically or upon events. This architecture is running regardless of whether the game is single-player or multi-player.

Torque implements the client-server model using a single executable. That is, whenever the engine is run, it contains both a server and a client. In the Torque implementation of a client-server architecture, the game world is controlled by the server. Game objects representing the game state are created and maintained on the server. For that reason, these objects are often called *server objects*.

During a game, individual clients are provided with copies of server objects. These copies are called *ghosts*. All game calculations are done using server objects. Thus, only server objects affect the game and its outcome. In order to keep clients up to date, the server will send information across the individual server-to-client connections, updating the ghost on each client.

Clients then render the game based on the state of their own ghosts. Figure 14 depicts the concept of server objects and copies of those objects being ghosted to a client. The server is aware of all server objects and all ghosts. Individual clients are only aware of their own ghosts. This provides a strong measure of security and prevents a number of game cheats involving clients having direct access to server objects and/or knowledge of other clients' objects.

Torque game engine games operate in any of three modes: dedicated-client, dedicated-server, and client-server.
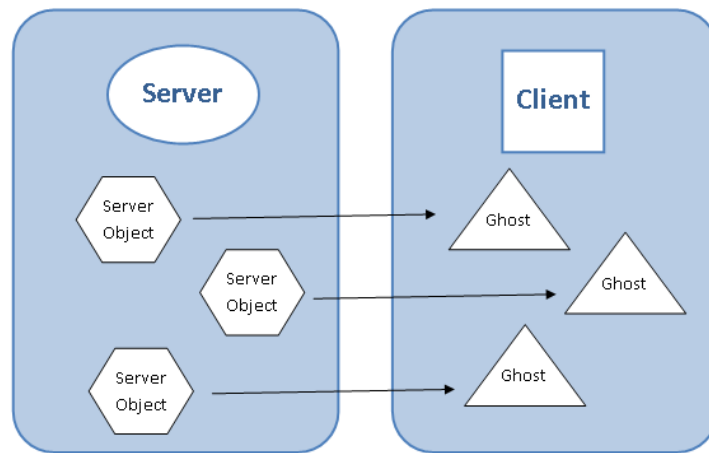
Fig. 14. Torque game engine server objects and ghosts.

In dedicated-client mode, the engine is only executing client tasks. It parses user inputs, sends these inputs to the server, and does any rendering necessary to present the game. To participate in a game, the engine must connect to another copy of the engine running in either dedicated-server or client-server mode.

In dedicated-server mode, the engine is only executing server tasks. It receives user inputs (from clients), maintains the game state, and updates clients regarding that state. External copies of the engine running in dedicated-client mode may connect to this server to participate in a multiplayer game.

In client-server mode, the engine is executing both client and server tasks. An engine running in this mode can be used to implement either a single-player game or a multiplayer game. Figure 15 shows the three modes of Torque.

(a) Dedicated-client mode   (b) Dedicated-server mode   (c) Client-server mode
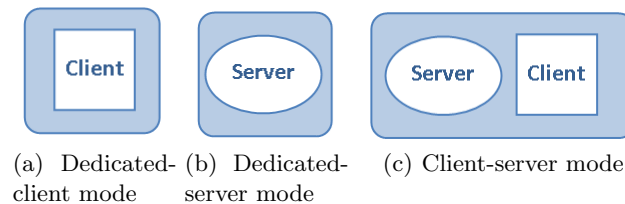
Fig. 15. Torque game engine modes.

Given these three modes, a game can be interconnected using one of the three connection schemes. The connection scheme we select is based on the game type we wish to run. Figure 16 shows a diagram of these three game types.

(a) Single-player connection (b) A multiplayer listen server connection (c) A multiplayer dedicated connection
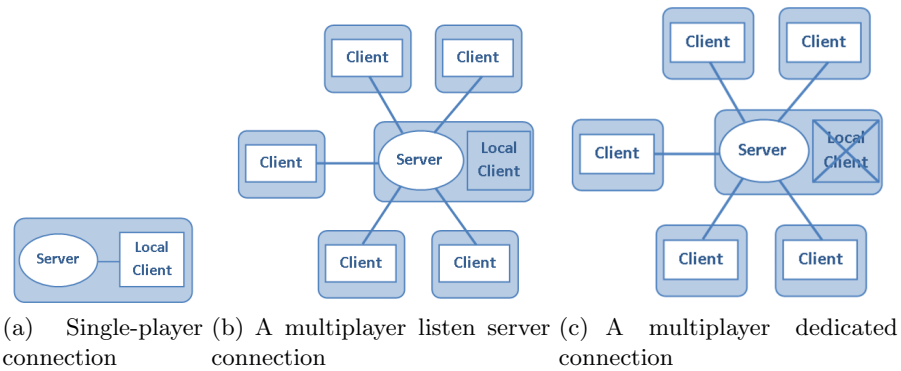
Fig. 16. Torque game engine client-server interconnection diagrams.

The simplest game type is the single-player game. This is accomplished by running a single instance of the executable on one machine. In this case, the server and client connect via an internal (local) connection. When this connection is requested, the server becomes active. The second game type involves a single executable with an active client and an active server running on one machine as a listen server. One player uses the local client and a local connection. The remaining players use client-only executables, running on separate machines, and connect remotely to the listen server. This mode is appropriate for local-area network parties and other cases where a user wants to host a game while participating. The last game type involves a single executable running as a dedicated server. This means that only the server is active. Multiple client-only executables, running on separate machines, can then connect with this executable, again allowing for multiplayer games. Although this could be used for a local-area network party, it is more suited to a professional hosting setup, where your company hosts one or more sessions on a machine used only as a server. This last game type was used in the implementation of the virtual fair. Figure 17 depicts an example of a three-person wide-area network party which uses six instances of the class *GameConnection*. This is, in fact, an example of the current settings for the virtual fair.

The Torque game engine is implemented in C++. It also allows the use of scripts written in its own scripting code similar to C++. The game engine also has its own integrated development environment (IDE) called *Torsion*. This facilitates the development of prototypes without the need for a constantly compilation. It also uses 3D graphics libraries, such as OpenGL and DirectX. The function of the server is to synchronize the position of clients and other animated objects and to detect possible collisions between them during the execution of a mission. In addition, it controls the messages sent among the clients and provides the infrastructure for the chat system. As mentioned above, currently the virtual fair consists of three scenarios: one for the common area, one for pavilion 1 and
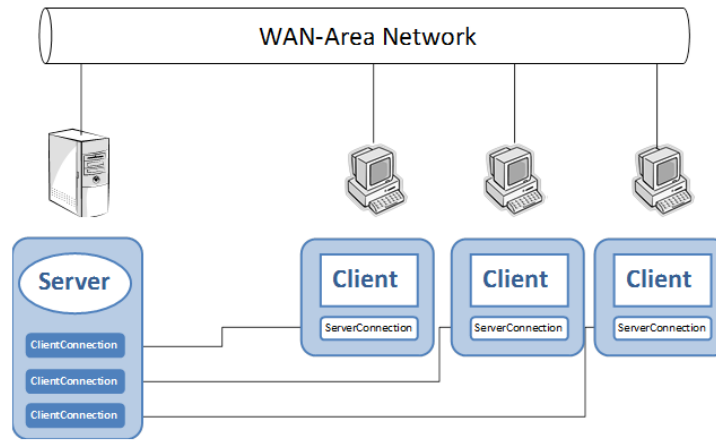
Fig. 17. Three-person wide-area network party.

one for pavilion 2. Figure 18 shows an example of connection between multiple clients and servers.

A single machine can run multiple instances of the game engine server. This allows us to have multiple missions running on a single computer, each one assigned to a different TCP/UDP port. In a current virtual fair setup this means that we need three server game instances running. However, each server can be migrated to a separate machine depending on system load, thus making this architecture scalable.
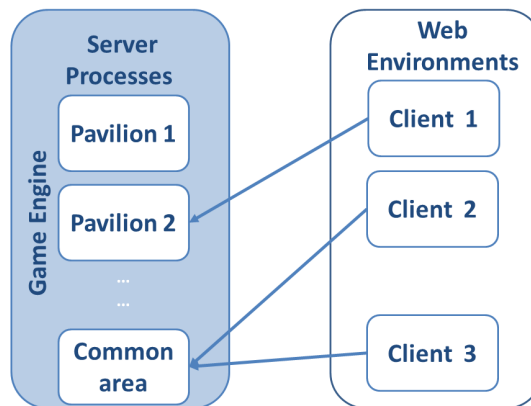
Fig. 18. The application organization of the client/server module.

Through the client-server module of the web environment, information is passed to the server game engine, including the username, avatar model, color combinations, position and rotation of the avatar in the world. An XML file is generated with this information, which is needed in order for the plug-in to run the client game engines, connecting them to the correct mission with the options selected by the user.

When a client connects to a game engine server for the first time, the web browser plug-in downloads a client module from the web server and installs it on the client machine. This client module contains the client version of the game engine and all the data needed to visualize the virtual world. This software is stored in multiple compressed and encrypted files. For subsequent connections, the files will not be downloaded again because they are already on the client machine. Even if an error occurs during the software download, the web browser plug-in is able to resume the download, reducing the data transferred between the client and the server.

8   Conclusions

In this paper, a multiuser 3D virtual world for business has been introduced (Figure 19). Firstly, we have taken advantage of the latest technology for the creation of 3D games. Secondly, the universality of the virtual fair was an important objective. To this end, we have used a client-server architecture on the Internet, a popular and widely used networked environment.

Fig. 19. This scene shows two users represented by avatars connected to the virtual world.

Our virtual fair offers an easy way for clients to connect. They just need a compatible Internet browser, which will connect to the virtual fail server using an plug-in. Once connected, the user can navigate though the world with the high realism offered by our solution. Moreover, we take into account client computer configurations and try to offer the maximum compatibility with standard graphics cards. Multiuser functionality is also present, offering companies the possibility of interacting or establishing communication with all the customers connected to the virtual fair. On the server side, fair content can be created with no specialized knowledge of graphics technology. That is, a virtual fair can be automatically created using a series of parameters such as the number of pavilions, stands and different style. In this way, the virtual fair administrator is able to easily design each stand and customize its content as desired. Finally, the virtual fair is generated and put online for the visitors.

Thus, our virtual fair is an interesting technology for companies and customers as it offers a highly realistic communication between them in a virtual world that is increasingly real. New functionalities may be incorporated into our virtual fair. Its architecture allows us to easily add new features by means of a plug-in system. In this sense, voice over IP support would be an important functionality to develop as it will enable us to offer better interaction between users. Thus, meeting or conference rooms, where conversations among the users could take place, would become essential in a virtual business world.

Moreover, support for video is another interesting area for exploration as the use of this technology to show new product features, company presentations, etc. would give to the customer a better view of the virtual fair content.

Acknowledgments

References

1. SecondLife: Retrieved January, 12, (2010) from http://secondlife.com/whatis
2. Guo, Y., Barnes, S.: Why people buy virtual items in virtual worlds with real money In SIGMIS Database, 38(4), pp. (69-76), New York, NY, USA: ACM, (2007)
3. Berger, H., Dittenbach, M., Merkl, D., Bogdanovych, A., Simoff, S., Sierra, C. : Playing the e-business game in 3D virtual worlds In ACM, Proceedings of the 18th Australia conference on Computer-Human Interaction, (pp. 333-336). New York, NY, USA: ACM.
4. Czerniawska, F., Potter, G.: Business in a Virtual World: Exploiting Information for Competitive Advantage. Ed. Palgrave Macmillan. (1998)
5. Greenfield, D.: Doing Business in the Virtual World. IT Infrastructure. Retrieved January, 14, (2010) from http://www.eweek.com/c/a/IT-Infrastructure/Doing-Business-in-the-Virtual-World/
6. Virtual Trade Fair Shows: Retrieved December, 14, (2010) from. http://www.virtualtradefair.org

7. TSSN, Onstream Media: 2010 Virtual Event Report: Insights & trends from Industry Insiders. Retrieved December, 20, (2010) from http://www.marketplace365.com/marketing/virtualtradeshowsurvey/TSNN.pdf

8. BusinessWomanFair: Retrieved January, 14, (2010) from http://www.businesswomanfair.com/information.htm

9. VirtualRealEstate Fair: Retrieved January, 14, (2009) from http://www.virtualrealestatefair.com/index_en.html

10. Luebke, D.: The present & future of Web3D. In Web3D '08: Proceedings of the 13th international symposium on 3D web technology, (pp. 6–6). (2008). Los Angeles, California: ACM.

11. Eberly, D. H.: 3D game engine design: a practical approach to real-time computer graphics. (2000). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. address.

12. Noh, S.S., Hong, S.D., Park, J.W.: Using a Game Engine Technique to Produce 3D Entertainment Contents. In ICAT '06: Proceedings of the 16th International Conference on Artificial Reality and Telexistence-Workshops, (pp. 246-251). (2006) Washington, DC, USA: IEEE Computer Society.

13. Trenholme, D., Smith, S. P.: Computer game engines for developing first-person virtual environments. Virtual Real, 12(3),181-187. (2008) London, UK: Springer-Verlag

14. Cencioni, R., Bertolo, S.: From Intelligent Content to Actionable Knowledge: Research Directions and Opportunities under Framework Programme 7. Retrieved January, (2009) from http://www.3dexpositions.com/

15. Groenewegen, S., Heinz, S., Frhlich, B., Huckauf, A.: Virtual world interfaces for special needs education based on props on a board, Computers&Graphics, 32(5), 589-596. (2008).

16. Maurina, E. F. III.: Multiplayer Gaming and Engine Coding for the Torque Game Engine. Natick, MA, USA: A. K. Peters, Ltd. Press.(2008)

17. Gehorsam, R.: The Coming Revolution in Massively Multiuser Persistent, Computer Journal, 36(4), (pp. 93-95), (2003). Los Alamitos, CA: IEEE Computer Society Press.

18. Belfore, L.A., Battula, S.: Virtual worlds: VRML clients linked through concurrent chat, In Proceedings of the 34th conference on Winter simulation, (pp. 518-524), (2002).

19. Holmberg, N., Wunsche, B., Tempero, E.: A framework for interactive web-based visualization. In AUIC '06: Proceedings of the 7th Australasian User interface conference, (pp. 137-144). Hobart, Australia: Australian Computer Society, Inc. (2006).

20. BusinessGlobal: Retrieved December, 14, (2010) from. http://www.businessglobal.com/

21. 3dexpositions: Retrieved January, 14, (2010) from. http://www.3dexpositions.com/

22. Torque 3D: Retrieved January, 12, (2010) from http://www.torquepowered.com/

23. Torque Constructor: Retrieved December, 12, (2010) from http://www.torquepowered.com/products/constructor

24. 3D Studio Max: Retrieved December, 20, (2010) from http://usa.autodesk.com

25. Indiepath Ltd.: Retrieved January, 12, (2010) from http://www.indiepath.com/