Interactive 3D Environments by using videogame engines

Roberto Andreoli Rosario De Chiara Ugo Erra Vittorio Scarano
ISISLab - Dipartimento di Informatica ed Applicazioni "R.M. Capocelli"
Università di Salerno
84081, Baronissi (Salerno), Italy
{dechiara,ugoerr,vitsca}@dia.unisa.it

Abstract

In this paper we study the technologies that are available for interactive and cooperative 3D environments. In particular, we describe how state-of-the art videogame 3D engines can be fruitfully used to create interactive 3D worlds that leverage on the photorealistic quality of the graphics and the high level of immersive navigation.

We provide, first, a categorization of videogame 3D engines from the point of view of their usage in creating interactive 3D worlds and show a comparison of the most important characteristics. Then, we show an example of how we used a commercial videogame engine to create an interactive an enjoyable visit to an archaeological site.

1 Introduction

One of the first ingenious employment of such technologies has been PSDoom [3] where Doom was used as interface to process management in Unix. The Doom interaction (a popular first-person shooter game) was used to "kill" or "nice" processes running in the machine.

By using Quake II 3D game engine, in [4] it is shown a 3D library where the user can browse books by looking in a 3D library (with bookshelves and so on). The weapon of the user was replaced by a laser pointer.

Another interesting application of videogame-like presentation techniques was studied in the "Mountain View" project, described in [1] where, by using an interesting combination of hyperbolic interface and videogame-based interaction, authors plan to provide a usable and efficient 3D interface to data archive.

Among the features that make attractive the videogame engines, there is the capability of obtaining cooperative environments that can enrich the quality of the experience of an interactive visit to a virtual world. For example, in [6] it is shown how to use games for cooperative virtual environments for architectural design. Another interesting feature that is used on rare occasions are the so-called Non-Playing Characters (NPC) as synthetic characters that can be programmed with a good degree of details and that allow to model scrupulously the interactions with the environment and the user, and with other processes outside the videogame (as, e.g., in [3]).

Our goal. We want to present a detailed comparison of the technologies that are available for creating interactive and cooperative 3D environments. We divide the technologies that are available in 3D engines mainly devoted to produce videogames and general-purpose 3D engines. Our thesis is that 3D videogames technology has, nowadays, reached a programmability level that allows a very efficient and effective deployment of virtual worlds based on these commercial products (by using the so-called "modifications" of the game (or *mods*)).

After a careful analysis and description of what is available, we present a proof-of-concept of the thesis presented in this paper, i.e., a reconstruction of an archaeological site that offers a very enjoyable and effective (i.e. informative) visit to the user. Our prototype allows an interaction with several characters as well as the ability of a cooperative visit to the site. The reconstruction objective was to obtain a reproduction that, besides providing the traditional 3D reconstruction of an ancient site, was aimed at an immersive visit that allows the user to be plugged in the middle of one of the most important ritual of that period, i.e., funerals in order to participate (in the true sense) to the funeral cortege and appreciate the importance of the act in that society. As a side effect, the user is brought to explore and interact freely with all the characters of the procession and within the surrounding, thereby obtaining a product that can be fruitfully used at any age and at any rate of interest, being appreciated even by teenager, given the videogame-like nature.

Organization of the paper. In Section 2 we review and compare existing videogame technologies with respect to their possible use in creating 3D worlds for educational purposes. In Section 3 we describe our example application to the reconstruction of an archaeological site and, finally, in Section 4 we conclude the paper with some final remarks.

2 Videogame 3D engines

In this section we give a review of the state of the art in 3D interactive environments that will be compared late, in the next section.

The designing of Virtual Environments (VEs) nowadays, can require the effort various different skills, not always directly related to computers, for instance it is quite widespread the use of architects to design the style of a VE or a scenographer from the cinema industry to better realize the background scene or general appeal of a videogame.

The point of view of content designer is to coordinate such various skills inside a coherent work. There is classical question to answer to :"Which framework is best suitable to develop an interactive 3d multi-user application ?"

The name usually given to this technological framework is "game engine", or *engine* for short.

An usual request from content designers is that the engine would work as high level layer over the graphics hardware in order that shade low level APIs like OpenGL or DirectX. Other typical requirements for the engine are that it should be scalable, robust to manage complex 3d objects, keyframe animations, interactivity.

2.1 A modern game architecture

A game engine is a complex set of modules that offers a coherent interface to the different functionalities that are usually more complex than the graphics rendering.

We provide here a short and not exhaustive list of these modules/functionalities, shortly discussing them. Figure 1 shows the structure of a modern game engine, with a succinct view of all the modules that made up the entire system. In the figure is also pointed out the entry level of the various typologies of developers that can use a game engine.

level 0: is the hardware that is both the computer and the network of the player. Modern videogames are usually structured as client-server application, we refer here to the client machine that is the player machine.

level 1: is made by the various engine which directly relate through the standard API (e.g. OpenGL or Direct3D for the graphics) to the hardware of level 0. At this level are defined the low level performance of a game engine, like the quality of graphics, the quality of sound and the network capabilities. The developer of a totally new game have to approach the development building from this level.

level 2: the game API is the layer of software which is in charge of abstracting the lower layers and implement functionalities like the AI of characters, the physics of the environment etc.... This is the layer which defines the high level characteristics of a game engine: how much realistic is the physics of the environment, or the behavior of the non playing characters in the game. Modern games are not always strongly directed to a certain type of game, so this level of the architecture usually provides a quite general API interface toward upper level. On top of level 2 work the developer who intend to develop a new game over a preexistent software architecture already functional.

level 3: this is the level where the gameplay is written. This is the level where the developers of a new game usually relate to. The presence, in the figure, of more than a modification means that, exploiting the low level engine general purpose vocation, is possible to develop various games whose "look'n'feel" is different but that share the same software core. The level 3 is the level which the *modders* relate to. A modder is a developer who creates a modification (since the jargon name) of an existing game, modifying maps, models, music, weapons etc...

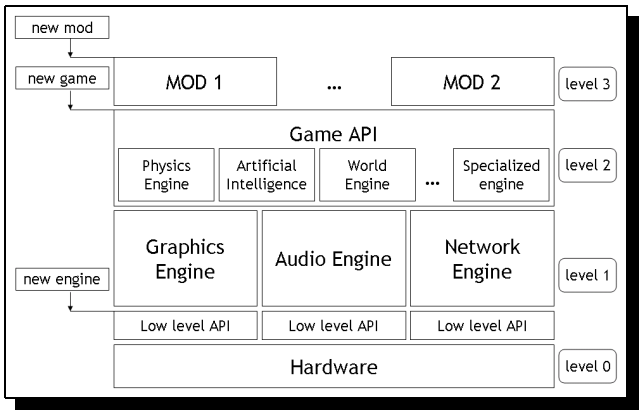

Figure 1: A modern game module schema. The first level is the entry level for mod creator (modder), the second level is the entry level for those who want to create a totally new game. The lowest level is the level on which a new engine is created from scratch.

2.2 The game engine in depth

A game engine is usually developed inside the project (and the funds) of a game that will hit the shelves for an

(hopefully) large audience. In this case engine name usually match the original game name, this is the case of the Quake Engine, the Half Life Engine and the Unreal2K3 Engine.

There are few engines written as general purpose software frameworks without a predetermined videogames title, but for a certain type of game; this is the case of engines like Renderware and CrystalSpace.

The general trend is that, mainly due to the major money investment, videogame inspired engines constitutes the state of the art of gaming and, more and more often, of field of research like real time computer graphics and finite element simulation. All this engines provide advanced features:

- *Graphics engine*. Nowadays, technological advances offer almost indistinguishable from reality real-time graphic rendering. The main part of research work has carried out in order to perform dynamic lighting and shadowing inside, even large, environments. Details can be easily added improving the visual appearance using effects like *bump mapping* and *displacement map* that adds information about the material an object is made of, without requesting complex geometric models. The use of programmable graphics card improves visual impact giving enabling the artists to inject visual improvements like specularity, reflection, refraction, or polished surfaces.

  Efficient data structures permit to manage large indoor/outdoor environments and procedural level-of-details over objects appearance. Particle systems can be used to render effects like beams, volumetric smoke, sparks. Image Post-Processing adds effects as depth of field, night vision, or motion blur.

- *Complex models*. Once that the rendering results are comparable with real world arose the need of developing also believable characters to give life to reconstructed worlds. Detailed and believable characters are usable. The models are capable of skeletal/bone system and animation blending in order to best mimicking the real behaviors of bodies. Facial expression, both lips moving and user following eyes, are recent features that can be used to perform language independent speech giving a more comfortable interaction with avatars.

- *Artificial intelligent*. Artificial intelligent is another core feature available thanks to increasing CPU resources. It can be used to produce desired behaviors to agents or non-player characters(NPC). The agents can walk, run, jump, crouch like any other human-driven player but can also be part of a squad of AI characters able to operate and coordinate together using sophisticated path finding algorithms, often borrowed from robotic researches. An agent can be a tourist guide, a teacher or a virtual presence useful to avoid to user feeling alone or loss. Decision-making triggers, can be disseminated around the environment to open a door, switch lights on, activate NPC behavior, or in general events.

- *World physics*. Basic physics, collision detection, rigid body are recent features available in nowadays engines. This features permit the design of a more responsive environment with realistic interactions with objects and also the creation of physically based vehicles like cars, trucks, helicopter etc. . .

- *Networking*. Modern games are, usually, network aware: the game experience provided on one's pc is usually transposed and emphasized in network game, both LAN and Internet. The on-line experience presents interesting social implications, visitors can choose to stay with their friends to, for instance, form a clan of warriors. Sightseeing tours can be organized and directed by a *Cicerone* (virtual or a special on-line user) using voice. Multiplayer environments could also cater some social desires like places where to meet with other users in ways similar to forums, chat rooms, message boards.

- *Sound engine*. Audio quality is a critical performance because it greatly improve the immersion in the synthesized world. A common feature is the 3d spatial sound, that gives and hint, like in real life, to the player about where the source of a sound is.

- *Tools*. An engine is a very complex and large framework, a set of tools are necessary to help people perform data entry and automate repetitive or error-prone tasks, improving reliability and efficiency and very important to save time. Usually engines offer a WYSIWYG world editor, terrain editor, and several adds-on to exports models from most common used 3d modelers. Editor allows artist to build virtual environments, then to place AI agents, triggers, and interactive objects. Furthermore they permit to extend game play using languages-driven scripts.

2.3 The comparison

In this subsection we intend to lay a small comparison considering a small number of nowadays videogame engines. The comparison among the environments described above will take into account several key aspects, in particular we consider what is the quality of results (e.g.

typology of effects) that can be achieved and at which development cost (e.g. quality of development tools) these results can be achieved.

In particular we will consider Half Life 2 as a paradigmatic example of nowadays game engine, Unreal Engine 3 as an example of what games engine can become and Torque which is a low-end low-budget engine.

Half Life 2. Half Life 2 (HL2) from Valve is the sequel of Half Life a famous game often tagged as "The best game ever". HL2 uses a game engine codenamed *Source* developed by Valve. Source is a deep modification of the Half Life engine that is itself an extension of Quake II Engine from id Software. HL2 hit the shelves on November 2004 and deliver up-to-date technology that has defined the new standards which future games have to face off. In particular there are some appealing functionalities that are also useful for general purpose environments: *facial animations*, NPCs in animated with Source have extremely realistic face expressions used to emphasize the sentences, perfectly pronounced with the proper labials. Another cutting edge feature of the engine is the physics engine. The use of physics engine within HL2 has changed the way of judging games, future games could not ignore the use of physics.

Unreal Engine 3. Unreal Engine 3 (UE3) is a complete game development framework for next-generation consoles and PCs, providing the vast array of core technologies, content creation tools, and support infrastructure required by top game developers. UE3 has the appealing feature of being cross platform, that means that game developed for PCs can be compiled and run on consoles at a small conversion cost. UE3 promises to be a new milestone in the realism, considering that it can boost to the max all the features available nowadays, having as a target hardware platform 2006 PCs.

Torque Engine. The Torque engine is a fully featured game engine with advanced multi-player network code, indoor/outdoor rendering engines, state of the art skeletal animation, drag and drop GUI creation, a built in world editor, and a C-like scripting language. A Torque engine license can be bought for $100. Unlike most commercial game engines, this engine is very cheap and is released with all source code. With UE3 and HL2 there is an important element to take into account before developing a new game, a developer can fully exploit pre-existent games structure (once again the mod architecture). The main drawback of Torque, and in general low-end low-price engine, is that the performances can never be compared with engines like HL2 and UE3.

X3D/VRML. X3D is an open standard for 3D content on the internet/web. It is intended to be the a format or language that is used for 3D interactive worlds on the web (replacing/extending the existing VRML97 standard). In X3D the world is displayed using a file format specification. The language is very robust and supports generalized 3D content for interactive web and other broadcast media. The main drawback is the lack of capable and flexible authoring tools that give the ability to create interactive worlds without any programming.

OpenGL Performer. [7] was the first commercially available scene graph for graphics. It provides a programming interface for developers who are creating real time visualization simulations or in general graphical applications. OpenGL Performer is written using OpenGL and the major benefit gained is that the development of real time applications avoid to being from scratch. OpenGL Performer is only oriented to high performance graphics, it is not capable to handle other important aspects of interactive environments like characters animations or physics. Ad-hoc extensions and tools must be written to support these features.

We will make this comparison among comparable technologies, that is, an game engine like Valve Source cannot be compared with OpenGL Performer or X3D/VRML, just because Valve Source is a complete game engine which includes all the modules of level 1 in Figure 1, while, for instance, OpenGL Performer is a graphics engine that, in a game development scenario, has to be used together with other specialized technologies in order to realize a complete game or interactive environment.

	Graphics engine	Complex Models	Artificial Intelligence	Physiscs Engine	Network Engine	Development Tools	Pricing Licensing
HL2	++	++	+++	+++	+++	+++	?
UR3	+++	+++	++	+++	+++	+++	?
Torque	+	+	+	+	++	+	+++

Figure 2: The table contains a comparison among three different game engine. The more '+' symbol is better. The '?' indicates we had no information.

3 A test case: a 3D reconstruction of an archaeological site

We describe the technical details of an application of videogame technology that is currently being designed [2]. The goal is to provide full access to a series of interesting painted tombs by simulating a virtual museum that is further enhanced with an interactive 3D reconstruction of a funeral cortege in the Andriuolo necropolis in Paestum.

3.1 The archaological site of Poseidonia-Paestum.

Poseidonia-Paestum is situated in the middle of the Gulf of Salerno, and its foundation can be traced back to the end of the 7th century b.C. by settlers coming from the town of Sibari. The site is well known for its temples, but also for its necropolises, that provided one of the very few examples of painted decorations that were well preserved up to current times. In fact, some tombs (probably of the richest people) are distinguished because of the paintings decorating the walls. Particularly rich is the necropolis in the territory of Andriuolo, situated north to the old inhabited area. The decorations show interesting details about the life of the dead and the rituals of his/her funeral. Currently, the whole corpus of the painted tombs is preserved in the National Archaeological Museum in Paestum but only some funerary paintings and a few objects furnishing the tombs can be exhibited in the Museum, because of lack of space and that makes interesting and useful a virtual reconstruction with a reenactment of the scene.

3.2 3D reconstruction guidelines

The 3D reconstruction is developed by using a state-of-the-art 3D engine directly from the videogame technology, Source engine from Valve, used for HalfLife 2 and recently released. The engine was recently released and allows to build virtual worlds (known as "mod" (as in modifications)) where the user is not only visiting a 3D virtual environment but he/she can also interact in first person with the synthesized characters that are acting in the scene.

Source is a first-person-shooter games engine that provides a photo-realistic 3d environment that enable a quick and effective development by a WYSIWYG World editor (Valve Hammer Editor). Source also provide Artificial Intelligence tools, such as Pathfinding and Decision Making algorithms that allow non-player characters (artificial representation of humans, animals, objects) to react with users' actions and with the other non-playing characters. The non-player characters can be also designed realistically since the engine provides skeletal animation, morphing, facial animation and animation blending so that the interaction is extremely shapable according to the designer needs.

In a sense, the main difference, from user's point of view, with traditional 3D scenes is simply that he/she is a character in a scene and plays with the environment and the characters. Of course, from the designer point of view, this environment offers exciting new avenues for providing content and information that are hidden and naturally provided to the user, e.g., in the form of a speech that is delivered directly from a character to the user if he/she approaches the character close enough.

The screenplay. Given the characteristics of interaction provided from Source engine, our design was created in the format of a traditional screenplay for a movie. In this context, we described, first, the scenario but also the characters, their dresses, the way they proceed in the funeral rite and how they can interact with the user/spectator.

The funeral takes place in early morning of a cloudy spring day in the fifth century b.C. The funeral procession comes from the town and the temples, that are seen in the background toward south.

The field where the procession enters is delimited by thick vegetation that only partially discloses glimpses of the sea and surrounding hills. The composition of the cortege follows the indications found in the painted tombs, assembled together for completeness.

After the leading flute player, the participants follow, bringing food and small vases (for perfumed oils). Young boys follow in with mourning women and animals to be sacrificed (an axe) led by a man. While these characters proceed, the user can walk around the cortege and interact briefly with each one, receiving some information on the nature of the dead person, or simple laments from the mourning women. The procession, then, proceeds to the burial site, where the painted tomb is already been placed. The burial is done by two men that take the body and place it the tomb with some ropes that are later pulled. Then, some vases are placed in the tomb and the tomb is closed. At the end, the procession moves back to town.

The user can interact with the characters, that can follow his/her movements in the procession and react with a plausible response to a series of repeated interactions with the user.

3.3 Project details

In particular [5] makes a comparison among Real Time Strategic (RTS) game engine and the First-Person Shooter (FPS) game engine.

Various points of interest exist in our project respect of traditional approach in the design of virtual reality. Virtual reality technologies are usually aimed to *reconstruct the architecture of a particular site*. Traditional virtual reality does not explain, for instance, life in the communities during a specific historical period; instead, it offers a static hyperrealistic reconstruction of a site, which does not distinguish between the actual remains and hypotheses. This is mainly due to the limitation of technologies used.

Our goal is deeply different, we intend to enable user to really participate to the rite, in fact, particular attention has been paid in modeling the semantic of the interaction between user and the characters present in the rite, and between the user and the architecture.

The scenario in which the rite is performed has been reconstructed using plausible textures for the foliage and trees, see figure 3.

Figure 3: Some screenshots took from the prototype we are developing.

4 Conclusions

In this paper, we advocate the usage of videogame engines in order to produce highly immersive, high quality 3d environments. Several are the key advantages with respect to other platforms. We believe that, at the expenses of buying a videogame engine, the quality of the result are greatly improved

Videogames do not only closely follow hardware evolution (graphic cards) but, in many cases, they push it and drive it with the overall objective of a highly realistic simulation. This happens, mainly, because of the large (economical) interest in one of the most successful field that, with the animation movies, drives an incredible amount of resources involved in the quest for ever more realistic products.

Just to give an example of the technological speed, the update frequency of drivers and technologies related to graphic cards is very high.

References

[1] Tim Altom, Melynda Buher, Michael Downey, and Anthony Faiola. Using 3D Landscapes to Navigate File Systems: The Mountain View Interface. In *Proc. of 8^th Int. Conference on Information Visualisation (IV 2004), 14-16 July 2004, London, England*, 2004.

[2] Roberto Andreoli, Rosario De Chiara, Ugo Erra, Vittorio Scarano, Angela Pontrandolfo, Luigia Rizzo, and Alfonso Santoriello. An interactive 3D reconstruction of a funeral in Andriuolo's Necropolis in Paestum. In *CAA 2005 - Computer Applications and Quantitative Methods in Archaeology*, 2005.

[3] D. Chao. Doom as an interface for process management. In *Proc. of SIGCGI conference on Human factors in computing systems. Seattle, Washington, 2001, USA*, 2001.

[4] Michael Christoffel and Bethina Schmitt. Accessing Libraries as Easy as a Game. In *Proc. of the 2^nd Int. Workshop on Visual Interfaces for Digital Libraries. July 2002*, 2002.

[5] Martin Meister and Martin Boss. On using state of the art computer games engines to visualize archaeological structures in interactive teaching and research. In *CAA 2003 - Computer Applications and Quantitative Methods in Archaeology*, 2003.

[6] Jules Moloney and Lawrence Harvey. Visualization and 'Auralization' of Architectural Design in a Game Engione based Collaborative Virtual Environment. In *Proc. of 8^th Int. Conference on Information Visualisation (IV 2004), 14-16 July 2004, London, England*, 2004.

[7] OpenGL Performer. *http://www.sgi.com/software/performer/*.