FROM ABSTRACT DATA MAPPING TO 3D PHOTOREALISM: UNDERSTANDING EMERGING INTERSECTIONS IN VISUALISATION PRACTICES AND TECHNIQUES

Visualization Research Unit, Birmingham Institute of Art and Design, 19 June 2007

Introduction to Visualization using Game Engines

Dr Nijad Al-Najdawi
Electronic and Electrical Engineering Department, Loughborough University, Leicestershire, UK, LE11 3TU
n.al-najdawi@lboro.ac.uk
Tel: +44 (0) 1509 635673
Fax:+44 (0) 1509 635231

Overview

Game engines started in the mid-1990s, especially in connection with 3D games such as first-person shooters (FPS). Such was the popularity of Doom (see Figures 1, 2) and Quake games that, rather than work from scratch, other developers licensed the core portions of the software and designed their own graphics, characters, weapons and levels [2].

Although the term was first used in the 1990s, there are a few earlier systems in the 1980s that are also considered to be game engines, such as Sierra's AGI and SCI systems, LucasArts' SCUMM system and Incentive Software's Freescape engine. However, unlike most modern game engines, these game engines were never used in any third-party products. Later games, such as Quake III Arena and Epic Games's 1998 Unreal were designed with this approach in mind, with the engine and content developed separately. At the very least, reusable engines make developing game sequels faster and easier, which is a valuable advantage in the competitive computer game industry [1].

Figure 1: the left hand side image is a screenshot for an object taken from Doom, and the right hand side of the image is a screenshot for an object taken from Doom 3

The continued refinement of game engines has created a strong separation between rendering, scripting, art-work, and level design. It is now common for example, for a typical game development team to have as many artists as actual programmers [2].

Introduction

A game engine is the central part component of a video game or other interactive application with real-time graphics. Game engines provide the fundamental technology, simplify development, and often enable the game to run on multiple platforms such as desktop computers and game consoles [1]. A game engine can be seen as an integrated collection of various modules that together run the video game. These modules include (see Figure3):

- A graphics module for 2D or 3D
- A physics module
- A collision detection module
- An input/output module
- A sound module
- An artificial intelligence module
- A network module
- A database module
- A Graphical User Interface module (GUI)

Different games will have some or all of the above modules. The development of Industry leading game engines requires huge amounts of time and resources. Many of these modules consist of advanced algorithms that in some cases have been developed for military applications, science, medicine or special effects for the film industry.

Figure 2: the left hand side image is “Skaarj” character from Unreal “” and the right hand side of the image is Berserker character from Unreal 3

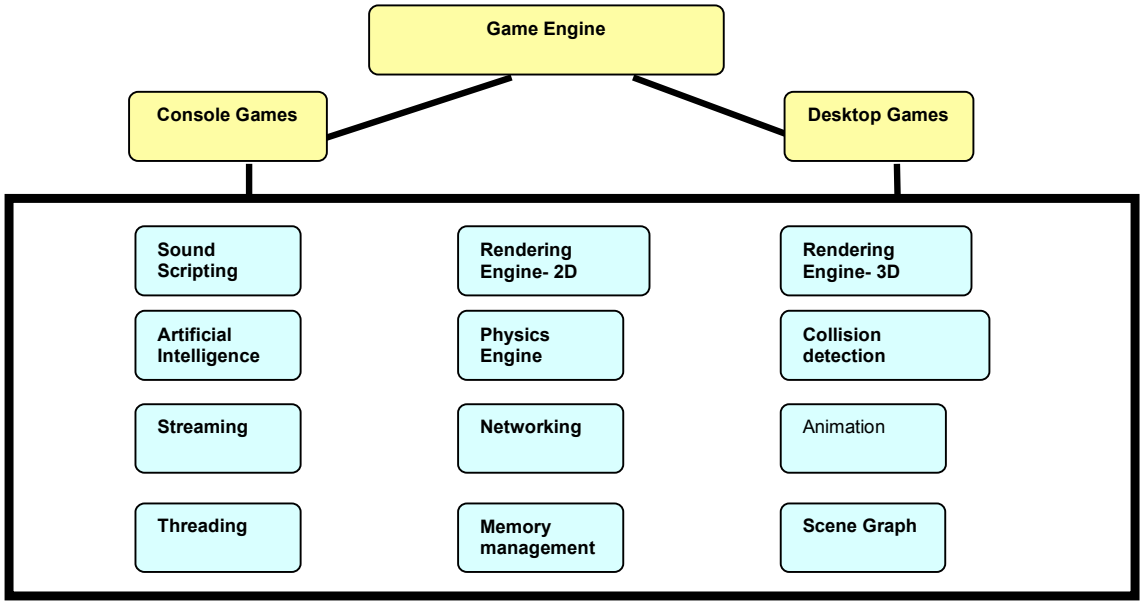

Figure 3: The core components of Game Engines

Game Engines

Game engines provide a suite of visual development tools in addition to reusable software components. In order to provide an integrated development environment to enable simplified, rapid development of games. Game engines usually provide platform abstraction, allowing the same game to be run on various platforms including game consoles and personal computers with few, if any, changes made to the game source code. They are designed with a component-based architecture that allows specific systems in the engine to be replaced or extended with more specialized components. Despite the specificity of the name, game engines are often used for other kinds of interactive applications with real-time graphical requirements such as marketing demos, architectural visualizations, training simulations, and modelling environments [4].

Threading has become a serious issue due to modern multi-core systems and increased demands in realism. Typical threads involve rendering, streaming, audio, and physics. Racing games have typically been at the forefront of threading with the physics engine running in a separate thread before other core sub-systems were moved, partly because rendering and related tasks only require updating at 30-60 Hz. For example, “Need For Speed” on the Sony Playstation ran its physics at 100 Hz as compared to Forza Motorsport 2 running its physics at 360 Hz [6].

Leading Free Game Engines [5]

- **Quake 3 - id Tech 3:** is a game engine developed by id Software for Quake III Arena and has been used in many games under the Quake III Arena engine and Quake III: Team Arena engine branding. id Tech 3 is a substantial improvement from the Quake engine and id Tech 2. Although id Tech 3 was derived from the id Tech 2, a large portion of code was new or re-written. It was succeeded by id Tech 4, which was derived from id Tech 3, but had none of the original code**.**
- **Crystal Space** is a framework for developing 3D applications written in C++. The first public release was on August 1997. Crystal Space is typically used as a game engine but the framework is more general and can be used for any kind of 3D visualization. Crystal Space is very portable free software, licensed under GNU Lesser General Public License.
- **OGRE :** (Object-Oriented Graphics Rendering Engine) is a scene-oriented, flexible 3D engine written in C++ designed to make it easier and more intuitive for developers to produce applications utilising hardware-accelerated 3D graphics. The class library abstracts all the details of using the underlying system libraries like Direct3D and OpenGL and provides an interface based on world objects and other intuitive classes.
- **Rage 3D** Free engine featuring Full Scene Management using scene graph, Several particle systems, Built in Bump-mapping, Character animation with weighted blending, Distance blending. supports full OpenGL 2.0 specifications.

Leading Low Price Engines [5]

- **Torque Game Engine Advanced 1.0:** as well as all the TGE 1.5 features this has atlas terrain generation editor, Torque GFX graphics layer, custom texturing materials, modern scene-graph and batched rendering engine - £150 for indie license.
- **Torque Game Engine 1.5 :**A big upgrade on 1.4.2, featuring Integrated Torque Lighting Kit, ShowTool Pro, Additional Art Assets and much more.
- **Torque Game Engine 1.4.2:** This old version of Torque still features a 3D toolset, geoterrain engine, networking engine and other features. Based on the technology used for Tribes 2.
- **TV3D SDK version 6.2:** This is a multi-language 3D engine, written primarily in C++ with support for Visual Basic 6, Visual Basic.Net, C#, Delphi, Python, and C++. Built upon the solid DirectX platform, you can quickly and easily prototype complex games and 3D applications in up to 80% less time.

Leading Premier Game Engines [5]

- **Doom 3 (idTech 4) :**Used for Doom 3, Quake 4, Prey and Enemy Territory: Quake Wars.
- **Unreal Engine 3:** Unreal Engine 3 is a complete game development framework for next-generation consoles and DirectX9-equipped PCs, providing the vast array of core technologies, content creation tools, and support infrastructure required by top game developers.
- **Unreal Engine 2X:** Unreal Engine 2X is the highly optimized engine behind the eye-popping visuals seen in Unreal Championship 2: The Liandri Conflict.
- **Unreal Engine 2:** Unreal Engine 2 is a complete game development framework targeted at today's mainstream PC's, Microsoft's Xbox game console, and Sony's PlayStation 2.
- **Gamebryo Element :** Gamebryo 2.3 takes development to a new level with a Scene Designer that allows developers to assemble new scenes, levels and worlds more quickly than ever.
- **Jupiter Ex :**Features a DirectX 9 renderer, Havok game dynamics, Havok Vehicel Kit, Networking engine, Game Database Editor (GDE) and content tools.
- **Havok :**Leading Physics engine technology. Havok Behaviour is a system for developing event-driven character behaviors in a game. Havok Physics is a State-of-the-art game physics solution, for use with in-house game animation systems. Havok Animation is a fast and flexible animation SDK and tool chain that provide optimized playback and real-time blending on current and next-generation game systems Havok FX™ is a special effects SDK and tool chain that leverages Shader Model 3.0 class GPUs to enable ten's of thousands of object collisions occurring in real-time, without putting additional burden on the CPU or otherwise slowing down the game.

Game engines vs. Virtual Reality engines.

Virtual reality (VR) is a technology which allows a user to interact with a computer-simulated environment. Most current virtual reality environments are primarily visual experiences, displayed either on a computer screen or through special or stereoscopic displays, but some simulations include additional sensory information, such as sound. Some advanced, haptic systems now include tactile information, generally known as force feedback, in medical and gaming applications. Users can interact with a virtual environment or a virtual artifact (VA) either through the use of standard input devices such as a keyboard and mouse, or through multimodal devices such as a wired glove, the Polhemus boom arm, and omnidirectional treadmill [7].

The simulated environment can be similar to the real world, for example, simulations for pilot or combat training, or it can differ significantly from reality, as in VR games. In practice, it is currently very difficult to create a high-fidelity virtual reality experience, due largely to technical limitations on processing power, image resolution

and communication bandwidth. However, those limitations are expected to eventually be overcome as processor, imaging and data communication technologies become more powerful and cost-effective over time [8].

The graphics displayed in the HMD will reach a point of near realism. The audio capabilities will move into a new realm of 3D sound. This refers to the addition of sound channels both above and below the individual. Within existing technological limits, sight and sound are the two senses which best lend themselves to high quality simulation. However, there are attempts being currently made to simulate smell. It is worth mentioning that simulating smells, while it can be done very realistically, requires costly research and development to make each odour, and the machine itself is expensive and specialized, using capsules tailor made for it. Thus far basic, and very strong smells such as burning rubber, cordite, gasoline fumes, and so-forth have been made. Something complex such as a food product or specific flower would be prohibitively expensive [7].

However, there are some conceptual differences when comparing a game engine to a VR-engine; the most important differences are [4]:

- Game engines have more visual effects and an easier workflow than the VR-engines.
- Game engines costs are generally much lower than VR-engines.
- The quality and usability of game engines exceeds that of the best Virtual Reality tools.
- The rapid development of game engines exceeds the virtual reality development.
- With game optimized multiplayer code it is now possible to create a presentation, and host a server online. Therefore, many people can then log on and experience not only the e.g. architectural presentation of a new building, but also the interaction of lots of people within this world.
- As the game engines and their tools mature, it is becoming easier and easier to make presentations.

Conclusion

Video game engines play a large role in the game development scene. A game engine is a framework used to create games from and is a complex engineering project that requires skill and dedication. The development of a game engine is not trivial, and the design and architecture will take time, resources, and teamwork. As video games become more complex, so does the engineering challenges that come up during development of a game. However, from the user point of view game engines have been widely accepted to replace the VR-engines when designing VR presentations and environment for the reason that they are have more visual effects and an easier

workflow than the VR-engines, moreover the game engines costs are generally much lower than VR-engines, the quality and usability of game engines exceeds that of the best Virtual Reality tools. Finally the rapid development of game engines exceeds the virtual reality development.

References

[1] Crawford, Chris (1982). The Art of Computer Game Design. http://www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html , accessed 8 October 2007

[2] Smuts, Aaron (2005). Are Video Games Art?, http://www.contempaesthetics.org/newvolume/pages/article.php?articleID=299 , accessed 6 October 2007.

[3] John Wills (2002). "Digital Dinosaurs and Artificial Life: Exploring the Culture of Nature in Computer and Video Games". Cultural Values (Journal for Cultural Research) 6 (4): 395–417.

[4] S. L. Kent, Engines And Engineering What to expect in the future of PC games, http://www.gamespy.com/futureofgaming/engines/ accessed: 15 October 2007

[5] 3D Game engine list, http://cg.cs.tu-berlin.de/~ki/engines.html, accessed 3 October 2007

[6] Sherman Chin Lit Kong – 3D Binary Space Partitioning (Game Engine Focus) - University Of Portsmouth.

[7] Burdea, G. and P. Coffet (2003). Virtual Reality Technology, Second Edition. Wiley-IEEE Press.

[8] Stanney, K. M. ed. (2002). Handbook of Virtual Environments: Design, Implementation, and Applications. Lawrence Erlbaum Associates, Inc., Mahwah, New Jersey