2010 Second International Conference on Games and Virtual Worlds for Serious Applications

An Engine Selection Methodology for High Fidelity Serious Games

Panagiotis Petridis, Ian Dunwell, Sara de Freitas, David Panzoli
Serious Games Institute
Coventry University Technology Park
Cheetah Road, Coventry, CV1 2TL
West Midlands, UK
Emails: {PPetridis, IDunwell , DPanzoli} @cad.coventry.ac.uk , Sara.deFreitas@coventry.ac.uk

Abstract**—**Serious games represent the state-of-the-art in the convergence of electronic gaming technologies with instructional design principles and pedagogies. Whilst the selection criteria for entertainment game engines are often transparent, due to the range of available platforms and engines an emerging challenge is the choice of platform for serious games, whose selection often has substantially different objectives and technical requirements depending upon context and usage. Additionally, the convergence of training simulations with serious gaming, made possible by increasing hardware rendering capacity, is enabling the creation of high-fidelity serious games which challenge existing design and instructional approaches. This paper highlights some of the differences between the technical requisites of high-fidelity serious and leisure games, and proposes a selection methodology based upon these emergent characteristics. The case study of part of a high-fidelity model of Ancient Rome is used to compare aspects of the four different game engines according to elements defined in the proposed methodology.

Keywords- high fidelity, game engines, serious games

I. INTRODUCTION

The convergence of high-fidelity simulators with serious games represents an area with increasing potential to fulfil cognitive learning requirements with a high degree of efficacy, whilst leveraging the advantages of game-based content to motivate and engage users. Modern games are frequently developed using engines which can be deployed on personal computers, game consoles, pocket PCs and mobile devices. The popularity of video games, especially among younger population, results in them frequently being perceived as an ideal medium for instructional programmes aimed at hard-to-reach audiences [1]. However, preliminary studies have also shown this demographic responds poorly to low-fidelity games [2], and as a result there has been a trend towards the development of more complex serious games that are informed by both pedagogic and game-play elements.

In the next section, we consider in greater depth the motivation behind the creation of high-fidelity serious games. Hence, through an analysis of background literature, we are able to highlight common pedagogic elements and hence recognise the subsequent technical implications. This leads us towards the selection methodology we propose in Section III.

II. BACKGROUND

Whilst it is true that the technical state-of-the-art in serious games mirrors that of leisure games [3], the technical requirements of serious games are frequently more diverse and wide-ranging than their entertainment counterparts. Serious game developers frequently resort to bespoke and proprietary development due to their unique requirements, such as the use of Blitz Games' in-house engine for developing the serious game: Triage Trainer [4]. This trend towards bespoke development highlights the difficulties that exist for game engine developers in accurately understanding and comprehensively supporting the needs of instructional designers.

Although many serious games have limited visual interactivity, immersion and fidelity, there is an increasing motivation to create serious games that intend to support situative (social and peer-driven) and experiential pedagogies; partially because behaviourist approaches have been shown to be limited (e.g. people learn to play the game, not address learning requirements), whilst cognitive approaches struggle to impart deeper learning in the areas of affect and motivation[5]. Furthermore, recent work by Mautone [6] demonstrated enhanced learning when introducing game elements to a standard flight simulator. Consequently, re-evaluation of simulator approaches to incorporate game and game-like elements places an increasing demand for serious game developers to deliver high-fidelity solutions. Given this motivation to create immersive, high-fidelity serious games, an obvious development choice is to utilise game engines which provide 'out of the box' support for state-of-the-art desktop GPU rendering and physics. In the remainder of this section, we discuss a range of key considerations when selecting the technology behind serious games, supporting effective pedagogy, the learner, and their context.

Fritch [7] and colleagues compared different games engines for large-scale visualization of outdoor environments focussing mainly on the issues of composability. Similarly Shiratuddin compared various game engines for visualizing large architectural scenes mainly focussing on accessibility and the availability of game engines [8]. Following the analysis of all these potential factors, we define our methodology to include areas, such audiovisual fidelity, functional fidelity, composability, availability and accessibility, networking, and heterogeneity.

A. Fidelity

High-fidelity in serious games is typically seen as desirable in situations where there is a need to transfer process knowledge learnt within the game to real world situations; and thus the closer the similarity between real and virtual space, the more effective the learning transfer is likely to be. Although a link between learning transfer and verisimilitude of learning activity has been observed particularly in training contexts [9-11]), this link does not necessarily hold true in all game-based learning scenarios; Jarvis and de Freitas [4] suggest that the level of fidelity required must be mapped onto learning objectives. Furthermore, an over-emphasis upon visual fidelity can mask the complexities of producing verifiable and replicable learning activities and experiences.

The need to engage the learner; and specifically, immersion through high-fidelity content is one such mechanism through which such engagement can be achieved. The concept of immersion is a common one in serious games, although the components that constitute an immersive experience can be more difficult to define. The capacity to immerse learners is a significant consideration, although the means for achieving this immersion can be diverse; from highly visual content to less technical approaches such as narrative immersion [12]. Robertson and colleagues [13] looked specifically at the relationship between user and standard desktop PC as an interface for virtual reality, and compared it to head-tracked systems. Robertson claims “immersion should not be equated with the use of head-mounted displays: mental and emotional immersion does take place, independent of visual or perceptual immersion”, an opinion reinforced by Csikszentmihalyi and Kubey [14]. Thus a further discretisation of the concept of immersion between psychological and perceptual levels is identified. The role consistently is one of ‘drawing in’ the user, such that they experience a perceptual shift between simply viewing the screen and existing within the environment. Breaks in consistency, such as those induced by low frame rate, or discontinuities in world content, are shown to have a significant negative impact on immersion [14].

The target demographic of the leisure game consists predominantly of users experienced with leisure games, and who thus expect a level of fidelity common to commercial entertainment games [2]. This is often a core concern when attempting to reach demographics unresponsive to other training programmes or educational material, and this objective is often at the core of the case behind the selection of a serious game as an educational platform.

With respect to all three of these aspects, there are a number of dimensions in which fidelity must be considered. At a high level, there are many aspects of a game that can be represented with differing levels of fidelity; the narrative, the depth of visual and auditory content, the interaction medium, and the behaviour of characters and objects within the game world. Whilst all these concerns must be reflected upon when designing a serious game, in terms of engine selection, a clear distinction exists between affordances for audiovisual and functional fidelity. It is possible to engineer a world which appears realistic but does not behave in a realistic fashion. Although increasing audiovisual fidelity often implies increased functional fidelity – as a virtual room fills out with furniture to become more visually realistic players start expecting that furniture to function as it would in the real world. For example, placing a virtual telephone on top of a desk can bring with it a host of potential questions from users expecting to be able to dial out.

B. Accessibility

Unlike leisure games, target demographics for serious games are often non-game players, with little interest in technology or knowledge of user interfaces. Furthermore, the developers of serious games may be instructional designers wishing to explore a new medium, rather than traditional game developers seeking to develop instructional content. Therefore, the capacity of the engine to support both developers and users with limited expertise is of relevance. Development of games such as Triage Trainer [4] has shown that conventional keyboard and mouse interaction in a world with multiple degrees of freedom can prove initially overwhelming for non-gamers. As such, serious games can often require interfaces that deviate from those common to entertainment games, and seek to simplify interactions based upon an understanding of learning requirements.

C. Networking

Multi-user elements are often specified at early-stage designs of serious games, since they often affect the nature of the game and its role within a training programme as a whole. User interaction within the game itself is often used to address the difficulties in automating the behaviour of non-player characters in a believable and coherent fashion. In this context, instructors play the role of virtual characters in order to converse and interact with learners in a realistic and adaptive manner. Whilst this can be an effective way of creating believable virtual scenarios, it suffers from limited scalability due to the availability of instructors, and practical limits on pupil-tutor ratios.

Support for larger-scale communities and social elements are gaining increasing recognition within the leisure gaming community as a mechanism for increasing uptake and long-term play. As an example, the recent Guitar Hero [35] game wraps a cognitively simple task within a socio-culturally motivating setting, and has consequently proved highly successful commercially. Within serious games, social elements often take the form of online communities, and the convergence of games with social networking technologies remains an area of interest.

D. Heterogeneity

Zyda identified as early as 2000 [15] three fundamental challenges in large scale multi-user virtual environment design, which he defined as composability, scalability and heterogeneity. These remain substantial challenges within current virtual environment research. With respect to serious games, heterogeneity is of particular concern, since target demographics are frequently ‘non-gamer’, and thus platforms capable of deployment across a wide range of hardware and software platforms are significantly advantageous. A serious game engine needs to track player behaviour and to assess their ability, capturing and reporting on those metrics. Depending on the game segment and market segment the game engine may need to be able to interact with real-time data such as GPS systems, simulators, and players and non-player characters.

E. Composability

Serious games often seek to model real world locales and situations, or adapt real world data for use in games with minimal development overheads. Composability in this context is used to describe both the reusability of content created within a game engine, and also its capability to import and use data from common or proprietary sources. If the scene consists of many high-fidelity objects at various distances, it may be possible to adopt a low level-of-detail approach [16] and use less complex geometry, or even dummy objects [17], to approximate distant objects [18]. Another issue that the designer/developer has to consider is when attempting to import a high-fidelity model onto a game engine. This can be achieved on a large-scale through an algorithmic approach that seeks to provide automated conversion between formats, and import the model as a whole into the game engine, although this relies upon the information and tools being available to perform the conversion effectively.

Algorithms intending to support such an approach face a number of challenges: decomposition of a mesh into multiple levels-of-detail is difficult to optimise since the perceived visual fidelity of the resultant lower-resolution meshes is linked to the perception characteristics of the user, and hence a solution is not only mathematically complex, but must also consider how humans perceive and integrate features of a three-dimensional scene [19]. Therefore, developers are commonly forced to select specific components of the model for conversion, and progressively convert and integrate them into the game engine by hand, a task necessary when conversion tools are inadequate or the original data format has insufficient information for the game engine; often the case when models are developed without adequate information on materials, textures, bump maps, or levels-of-detail.

III. Requirements and Features of Games Engines

In order to form our selection methodology, we reflect upon the considerations discussed previously in Section II, alongside a lower-level technical analysis of current engines.

A. Summary of Requirements for Serious Games

In short, we define key elements arising from the background as fidelity subdivided into visual and functional fidelity, consistency, (technical features in this area would include need to load between areas or stream geometry, whether the engine is standalone or web based – basically things that impact on immersion), and support for other tools for creating immersion and flow such as narrative (corresponding features include: game scripting tools), particularly narrative which supports non-linearity (artificial intelligence, artificial life). This ties into immersion; we deliberately avoid "immersion" within the methodology and focus on the elements that contribute to immersive experiences.

Additional technical elements include heterogeneity (which platforms can the engine be deployed on; what hardware requirements; can it scale automatically), accessibility (support for non-standard interfaces and devices; as well as support for standardised interfaces (e.g. WASD), and multiuser support, beneficial since in the absence of sophisticated AI, human instructors often play a role in virtual learning experiences, and similarly socio-cultural elements can be key motivators as mentioned above.

TABLE I. Methodology for comparing engines in SG

Audiovisual Fidelity	*Rendering*
	Animation
	Sound
Functional Fidelity	*Scripting*
	Supported AI Techniques
	Physics
Composability	*Import/ Export Content*
	Developer Toolkits
Accessibility	*Learning Curve*
	Documentation and Support
	Licensing
	Cost
Networking	*Client Server/ Peer–to- peer*
Heterogeneity	*Multiplatform Support*

B. Overview of Modern Game Engines

The game engine is responsible for the rendering, the physics, the Artificial Intelligence and other game mechanics [7].

Game engines should be distinguished from graphics engines that come only with rendering capabilities, and also from Software Developer Kits (SDKs) that aid game development. The reason for this differentiation is that graphics engines impose limitations as to what and how things can be included in a game, whilst SDKs are much more flexible but with narrower focus. For example, Gamebryo is a very flexible proprietary renderer but has no collision detection or physics capabilities, unlike Havoc which is solely a physics engine. Similar middleware include Criterion's Renderware and Speedtree.

Game engines are commonly implemented as an integrated collection of modules, with differing functionalities. Analysis of a range of current game engines including Quest3D, Unreal, CryEngine suggests the following modules are common:

1. Graphics Module
2. Physics Module
3. Collision Detection Module
4. I/O Module
5. Sound Module

6. AI Module
7. Network Module

The graphics module is responsible for the generation of the 2D/3D graphics on the environment, including libraries for texture mapping, shadowing, lighting, shader effects, etc. The physics module ensures that objects behave according to physical laws, for example objects fall under gravity and glass breaks. The collision detection module is used to ensure that certain actions will occur when two objects collide. Furthermore the input/output module is responsible for the input and output device which can be integrated into the 3D engine.

One of the most important elements of the creation of serious games is the visual representation of these environments. Although serious games have design goals that are different from those of pure entertainment video games, they can still make use of the wide variety of graphical features and effects that have been developed in recent years. Most game engines provide support for texture mapping, shadowing, lighting and shader effects, in their graphics model. Additionally, modern engines frequently include a selection of such effects, which can include more traditional image processing, such as colour correction, film-grain, glow, or edge-enhancement, as well as techniques that require additional scene information, such as depth of field, motion blur [17].

An Artificial Intelligence module, often used to create objects or "Non Playing Characters" (NPCs), is able to interact "intelligently" with the player. An important aspect in the creation of realistic scenes is to create in the game environment intelligent behaviours for the inhabitants of the virtual world, which is achieved using Artificial Intelligence (AI) techniques. However it is important to understand that when we refer to the AI of virtual entities in game engines, that which we refer to is not truly AI – at least, not in the conventional sense [20]. The techniques applied to computer games are usually a mixture of AI-related methods whose main concern is the creation of a believable illusion of intelligence [21], e.g. the behaviour of virtual entities only needs to be believable enough to convey the presence of intelligence and to immerse the human participant in the virtual world.

Additionally in order to increase the realism of the scene, game engines incorporate either 2- or 3-dimensional sound modules in order to increase the sense of immersion. Several results are appearing in the literature concerning the interaction between different modalities, such as vision and audition, to achieve a higher sense of presence, or on the role of reproduction quality which allows the user to enhance their perception of immersion [22, 23].

A network module is responsible for the multiplayer implementation of the game. As a result, players could cooperate in exploring an area, or exchange opinions about certain aspects of a virtual environment while being located in different areas of the world.

The IO module provides support for different input/output devices. This module provides tools that allow the user to communicate and interact with the game environment. Most game engines provide support for standard input devices such as, joysticks, gamepads, keyboard. Technological improvements and cost reduction in computing power, display and sensor technology have resulted in a widespread use of 3D Input Devices [24-26]. Such devices such as the Nintendo Wii and Playstation sixaxis controller provide 6 Degrees Of Freedom interaction that could enhance the user interaction with the environment and increase the immersion of the user.

IV. Defining an Engine Selection Methodology

We used an accurate high-fidelity model of the Colloseum, taken from the Rome Reborn project. This project represents the world's largest digitisation project, and has been running for 15 years. The main aims of the project are to produce a high resolution version of Rome 320AD (see Figure 1), a lower resolution model for creating a mash-up application with Google Maps, and finally the collaborative mode of the model for use with virtual world applications and aimed primarily at educational purposes [27, 28].

The Rome Reborn model was initially provided to us in 3D Studio Max (3DS) format, and consists of 100 million polygons. The model includes a digital terrain map with the hills, valleys, and water features of the city. Additionally it is composed of 7000 buildings within the late-antique Aurelian Walls [27]. For this evaluation study we will use the Colloseum as a test-case in evaluating the methodology using four selected game engines, chosen for popularity. The Colloseum consists of 520k polygon and the textures originally in RGB format.

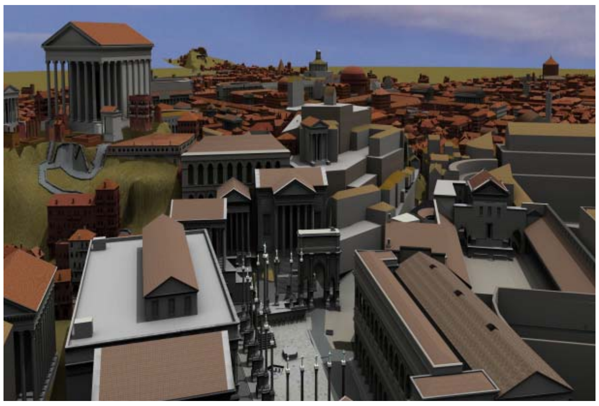

Figure 1: Rendered image from the Rome Reborn dataset[2]

We imported sections of the model onto the most popular engines to analyse emergent problems against the parameters of the model. A predominant issue with the high-polygon Rome Reborn model in its original format is the load placed on the renderer, and subsequently the non-interactive frame rate that arises, due to the size of the model. Other projects working with this model have explored parallel rendering [28] techniques using global illumination, although these are currently unsuited to a serious game deployed on a typical entertainment platform, such as a desktop PC. However,

[2] http://www.romereborn.virginia.edu/gallery.php

modern game engines employ a range of strategies for supporting expansive environments containing large volumes of geometric data, such as multiple levels-of-detail [29] and occlusion culling [30].

Hence, the major challenge faced is in converting the model to such a format that the optimization methods employed by the game engine can be fully-realized. This highlights a major limitation of many game engines with respect to composability that content is often required in proprietary formats and conversion between model formats requires additional work by-hand to address conversion errors, which can represent a significant volume of work for larger models, Through comparing game engines such as this study it is envisaged that standardised approaches to solve this problem will be incrementally developed.

A. Comparison of the Selected Game Engines

Quest3D is a very flexible authoring environment for real-time 3D applications. The edit-while-executing and graphical nature of Quest3D makes it one of the most intuitive tools to work with. Quest3D is used by developers, educational institutions and VR companies [31]. Projects include anything from small freeware games to huge industrial virtual reality setups. The second engine under comparison is the Blender engine [32] which is an open source 3D modelling, rendering, animation and real-time 3D game/simulation development system. The third engine is the Unreal Engine 3 [33], which is a complete game development methodology for next-generation consoles and DirectX9-equipped PC's, providing the vast range of core technologies, content creation tools, and support infrastructure required by top game developers. The final engine under comparison is the Unity engine [34], which is a game development tool that allows the developer to create games for different platforms such as the iPhone, Nintendo Wii, Mac and PC. The game engines were compared according to the elements proposed in the methodology.

Although the benefits of high levels of audiovisual fidelity in serious games apply particularly to games wishing to provide learning though simulation, high visual fidelity also contributes to a number of other factors, such as the initial uptake and motivation amongst learners; a study by de Freitas and colleagues [2, 4] suggests that experienced gamers fail to engage with serious games which are perceived to be of significantly lower-fidelity than leisure games with which they are already familiar. Quest3D, Unity and the Unreal engines provide high-fidelity standards and provide support for all the latest technologies of computer graphics as can be seen from Table II. Using these features such texturing, lighting, shadows and special effects the developer has the potential to deliver the same perceptual quality as the user was present in the real scene. Thus the player can feel actually present in the real scene being depicted.

TABLE II. AUDIOVISUAL FIDELITY

		Quest3D	***Blender***	***Unreal***	***Unity***
Rendering	***Texturing***	Basic, Multi-texturing, Bump mapping	Basic, Multi-texturing, Bump mapping	Basic, Multi-texturing, Bump mapping, Procedural	Basic, Bumpmapping, Procedural
	Lighting	Per-vertex, Per-pixel, Lightmapping, Gloss map	Per-vertex, Per Pixel	Per-vertex, Per-pixel, Gloss/ Specular Mapping Light mapping	Per-vertex, Per Pixel
	Shadows	Shadow Mapping, Projected planar, Shadow Volume	Shadow Mapping,	Shadow Mapping, Projected, Shadow Volume	Projected planar
	Special Effects	Environmental Mapping	Environmental Mapping	Environmental Mapping	Environmental Mapping
		Particle Systems,	Particle Systems	Particle Systems	Particle Systems,
		Bill Boarding	Bill Boarding	Bill Boarding	Bill Boarding
		Lens Flares	Lens Flares	Lens Flares	Lens Flares
Animation		Forward Kinematics, Keyframe Animation, Skeletal Animation, Morphing, Animation Blending	Non-linear animation mixer with automated walkcycles along paths	Forward Kinematics, Keyframe Animation, Skeletal Animation, Morphing, Animation Blending	Forward Kinematics Keyframe Animation, Skeletal Animation Morphing, Animation Blending
Sound		2D Sound, 3D Sound, Streaming Sound:	2D Sound, 3D Sound	2D Sound, 3D Sound, Streaming Sound	2D Sound, 3D Sound, Streaming Sound:

Another major challenge in the selection of game engines for serious games is functional fidelity. Functional fidelity is closely related to the AI, physics and scripting. All the selected game engines provide support for various AI techniques, such collision detection, path finding. Additionally, all game engines have integrated their own physics engines, and furthermore, each of the selected game engines has support for scripting languages.

TABLE III. FUNCTIONAL FIDELITY

		Quest3D	***Blender***	***Unreal***	***Unity***
Scripting	Script	Yes	Yes	Yes	Yes
	Object Model	Yes	No	No	No
Supported AI Techniques	Collision Detection	Yes	Yes	Yes	Yes
	Path Finding	Yes	No	Yes	Yes
	Decision Making	Yes	No	Yes	Yes
Physics	Basic Physics	Yes	Yes	Yes	Yes

	rigid body	Yes	Yes	Yes	Yes
	vehicle dynamics	Yes	Yes	Yes	Yes

The reusability of content created within a game engine and the capability of importing and using data from common sources should be considered before selecting a game engine. Additionally in serious games the developers need to have access to the SDK, GDK in order to add different devices (e.g. Neurosky – a brain-computer interface device), or connect the game engine with learning management systems or with other software APIs. Table IV compares the selected engines according to the suggested categories.

TABLE IV. COMPOSABILITY

	Metrics	*Quest3D*	*Blender*	*Unreal*	*Unity*
Import/ Export Content	CAD Platforms supported	3ds max, maya	Build-in CAD	3ds max, maya,	3ds max, maya,
	Import Export Limitations	No	No	Yes	No
	Content Availability	Small	Large	Large	Medium
Developer Toolkits	SDK/GDK	Yes	Yes	Yes	No

Another major challenge in the selection of game engine for serious games is the accessibility (see Table V.).

TABLE V. ACCESSIBILITY

	Metrics	*Quest3D*	*Blender*	*Unreal*	*Unity*
Learning Curve		Steep Learning Curve	Medium	Medium	Medium
Docs and Support	Docs Quality	Docs and tutorial	Docs and tutorial	Docs and tutorials	Docs and Tutorials
	Technic al Support	Yes	No	Yes	Yes
	Commu nity Support	Yes	Yes	Yes	Yes
Licensing		Creative, Power and VR Editions	Open Source	Commercial Licence.	Indie and Pro Version
Cost		Cost is between 1249, 2499, 9999 euros	free	free for educational purposes	149 and 1099 Euros respectively

The next step in the selection process of the game engine is to focus on the heterogeneity of the engines and their network support (see Table VI). All the selected engines support Client-Server architectures. However if the serious game is going to support a large virtual world with hundred of users, a network supported layer has to be built (this would be the case for Unreal Runtime and Blender). From the selected engines only Unreal engine and Unity are multiplatform.

TABLE VI. NETWORKING AND HETEROGENEITY

		Quest3D	*Blender*	*Unreal*	*Unity*
Networki ng	*Client-Server,*	Yes	Yes	Yes	Yes
	Peer-to-Peer	Yes	No	No	No
Heter ogene	*Multiplatform*	*No (PC version only)*	*No (PC version only)*	*Yes*	*Yes*

V. DISCUSSION

By comparing the characteristics of the selected game engines (See Figure 2) we can observe the following observations.

The Unreal 3 engine outperforms all other engines included within the comparison, in terms of audio and visual fidelity, functional fidelity and composability. This engine can therefore be used as a benchmark in order to compare different game engines in serious games. However, the main disadvantage of this game engine is its limited accessibility, a consequence of the high costs associated with the engine.

Quest3D and Unity are very similar with respect to visual and audio, as well as functional, fidelity (refer to table 3 and table 4). Both engines provide support for high resolution and realistic images that are used as textures for the 3D objects in the game's VE. Both engines features support for 24-bit, 32-bit and 64-bit colour images, multilayered texturing, volumetric environmental effects, shaders, advanced dynamic lightings and dynamic shadows.

However, Quest3D outperforms the Unity engine in its networking capabilities because it supports more networking protocols. However a major advantage of the unity engine is the multiplatform support (Unity is available for Macs, PCs, Nintendo Wii and iPhone).

In terms of accessibility all the selected engines have high documentation quality and all the engines with the exception of Blender provide technical support. In terms of community support, Unity's community is larger than that of Quest3D (refer to tables 4 and 5) and a variety of artists and game developers are using Unity.

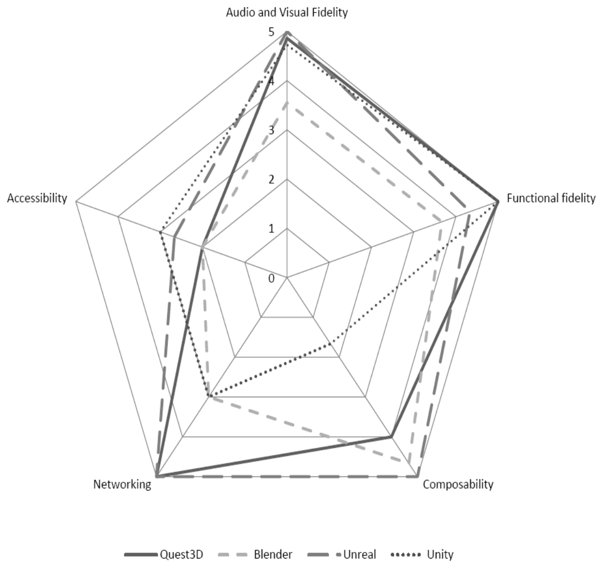

Figure 2: Comparison of the selected game Engines

The Blender game engine is an open source project; however it was outperformed in every category by Quest3D, the Unreal 3 engine and Unity.

However, from the comparison of the game engines we identified that importing a 3D model from CAD software into the supported format of the game engine is a major issue and requires the developer to select specific components of the model for conversion and progressively convert and integrate them into the game engine by hand. Once this obstacle has been overcome, the game engines can create beautiful indoor and outdoor scenes occupied with non-player characters in real-time.

To utilise our methodology, the serious game developer needs to adopt a process wherein they relate learning requirements to game features, and subsequently map these features against the five categories described within this paper. This would enable them to accurately relate the technical features necessitated by pedagogy and instructional design, to the features available in current game engines. In terms of our Rome Reborn case study, we intend to investigate the efficacy of large-scale, high-fidelity worlds for learning, exploration, re-enactment and research, on both cultural and archaeological levels. In particular, the case study aims to investigate the suitability of using game technologies in order to support the archaeological exploration of historically accurate representation of ancient Rome, alongside the integration of different technologies. In order to achieve these goals, future development and extensions to current game engines must address all five of the key dimensions specified by our methodology.

VI. CONCLUSIONS

The creation of a serious game is a complex engineering project that requires skill and dedication. The development of a serious game engine is a complicated process that requires time, resources and teamwork. As serious games become more complex, so do the engineering challenges that arise during development of the game. Hence the selection of an ideal engine for this development is crucial. This paper proposed an evaluation methodology that will allow the developer to select the ideal engine based on the technical requirements of the serious game, which an associated link to the driving pedagogic factors. The proposed methodology is regarded as a starting point for a wider project towards overcoming particular issues with respect to composability.

VII. FUTURE WORK

Whilst our methodology relates overarching technical requirements to a range of modern engines, more research, testing and validation must still be done to relate learning requirements and instructional design principles to these technical features, a fact we have discussed in Section II. Ultimately, the design and implementation of effective serious games must be grounded in pedagogy, as well as technology, and therefore future work should address the many issues surrounding the equation of learning requirements to technical features.

Furthermore, analysis of the impact of the various engines on learners may offer be a potential avenue for future study. On a final note, whilst we sought to identify and evaluate the most popular and widespread game engines within the context of our framework, given the rapid evolution of technology in this area, comparison of additional and emerging game engines may also offer opportunities to further refine and validate the methodology.

REFERENCES

[1] Malone, T.W., Lepper, M. R., *Making learning fun:A taxonomy of intrinsic motivations for learning*, in *In Aptitude, learning and instruction: III. Conative and affective process analyses*, F.J. Snow R. E., Editor. 1987: Erlbaum. p. 223-253.

[2] de Freitas, S., Liarokapis, F., Rebolledo-Mendez, G., Magoulas, G., Poulovassilis, A., *Developing an evaluation methodology for immersive learning experiences in a virtual world*, in *Proceedings of 1st IEEE conference on Games and Virtual Worlds for Serious Applications*. 2009. p. 43-50.

[3] Anderson, E.F., McLoughlin, L., Liarokapis, F., Peters, C., Petridis, P., de Freitas,S. *Serious Games in Cultural Heritage*. in *10th VAST International Symposium on Virtual Reality, Archaeology and Cultural Heritage (VAST '09), VAST-STAR, Short and Project Proceedings, Eurographics*. 2009. Malta, 22-25

[4] Jarvis, S., de Freitas, S., *Evaluation of an Immersive Learning Programme to support Triage Training*, in *Proceedings. of the 1st IEEE International Conference in Games and Virtual Worlds for Serious Applications, IEEE Computer Society,*. 2009: Coventry,UK. p. 117-122.

[5] Egenfeldt-Nielsen, S., *Beyond edutainment: exploring the educational potential of computer games*. 2005, IT-University Copenhagen.

[6] Mautone, T., Spiker, V., Karp, D., *Using Serious Game Technology to Improve Aircrew Training*, in *In Proc. of I/ITSEC 2008*. 2008.

[7] Fritsch, D.a.K., M.,, *Visualization Using Game Engines*, in *ISPRS commission 5*. 2004: Istanbul, Turkey,. p. 621-625.

[8] Shiratuddin, M., Fletcher, D., *Utilizing 3D Games Development Tool for Architectural Design in a Virtual Environment*, in *7th International Conference on Construction Applications of Virtual Reality*. 2007.

[9] Park, G.D., Allen, R. Wade, Rosenthal, Theodore J., Fiorentino, Dary *Training Effectiveness: How Does Driving Simulator Fidelity Influence Driver Performance?* , in *Human Factors and Ergonomics Society Annual Meeting Proceedings, Training*. 2005. p. 2201-2205.

[10] Davidovitch, L., Parush, A.; Shtub, A. , *The Impact of Functional Fidelity in Simulator-based Learning of Project Management.* International Journal of Engineering Education, 2009. **25**(2): p. 333-340.

[11] Janet L. Grady, R.G.K., Carole E. Trusty,Eileen B. Entin, Elliot E. Entin, Tad T. Brunye, *Learning Nursing Procedures: The Influence of Simulator Fidelity and Student Gender on Teaching Effectiveness.* Journal of Nursing Education, 2008. **47**(9).

[12] Mott, B.W., McQuiggan, S.W., Lee, S., Lee, S. Y, Lester, J. C., , *Narrative-Centered Environments for Guided Exploratory Learning*, in *Proceedings of the Agent Based Systems for Human Learning Workshop at the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (ABSHL-2006)*. 2006: Hakodate, Japan.

[13] Robertson, G., Czerwinski, M., Van Dantzich, M. , *Immersion in desktop virtual reality*, in *Proceedings of the 10th annual ACM symposium on User interface software and technology*. 1997, ACM Press. p. 11-19.

[14] Csikszentmihalyi, M., Kubey, R., *Television and the rest of life: A systematic comparison of subjective experience.* Public Opinion Quarterly, 1981. **45**(3): p. 17-328.

[15] Zyda, M., Singhal, S., *Networked Virtual Environments: Design and Implementation*. 2000, ACM Press: New York.

[16] Engel, W., Hoxley, J., Kornmann, R., Suni Zink J., *Programming vertex, geometry, and pixel shaders. line book available at: http://wiki.gamedev.net/, 2008. 9, 11.*

[17] Akenine-Moller T., H., E., Hoffman,N., *Real-Time Rendering*. 2008, A. K. Peters.

[18] Sander, P.V., Mitchell, J. L. , *Out-of-core rendering of large meshes with progressive buffers*, in *ACM SIGGRAPH 2006: Proceedings of the conference on SIGGRAPH 2006 course notes (2006)*. 2006. p. 1-18.

[19] Treisman, A., Gelade, G., *A feature-integration theory of attention.* Cognitive Psychology, 1980. **12**(1): p. 97-136.

[20] McCarthy, J. (2007) *What is Artificial Intelligence, Available At http://www-formal.stanford.edu/jmc/whatisai/whatisai.html.*

[21] Scott, B., *The Illusion of Intelligence*, in *AI Game Programming Wisdom*. 2002, Charles River Media. p. 16-20.

[22] Storms, R., Zyda, M., *Interactions in perceived quality of auditory-visual display.* Presence, 2000. **9**(6): p. 557-580.

[23] Freeman, J., Lessiter, J. , *Here, there and everywhere: The effect of multichannel audio on presence*, in *Proceedings ICAD,2001*. 2001.

[24] Petridis, P., White, M., Mourkousis, N., Liarokapis, F., Sifiniotis, M. Basu, A., Gatzidis, C. *Exploring and Interacting with Virtual Museums*. in *CAA 2005: The World in your eyes*. 2005. Tomar,Portugal.

[25] Mourkoussis, N., Mania, K.,Petridis, P.,White, M., Rivera, F., Pletinckx, D. . *An analysis of the effect of technological fidelity on perceptual fidelity*. in *To appear in proceedings of the IEA 2006 (International Ergonomics Association), 16th World Congress on Ergonomics*. 2006. The Hague, Netherlands.

[26] Fröhlich, B., Plate J., Wind J., Wesche G., Gobel M., *Cubic-Mouse-Based Interaction in Virtual Environments.* IEEE Computer Graphics and Applications, 2000. **20**(4): p. 12-15.

[27] Frischer, B., D. Abernathy, F.C. Giuliani, R. Scott, Ziemssen, H., *"A New Digital Model of the Roman Forum," in "Imaging Ancient Rome. Documentation-Visualization-Imagination.* Journal of Roman Archaeology, Supplementary Series 2006. **61**: p. 163-182.

[28] Frischer, B., *The Rome Reborn Project. How Technology is helping us to study history.* OpEd. University of Virginia, 2008.

[29] Hoppe, H. *Progressive Meshes*. in *Proceedings of SIGGRAPH'96*. 1996.

[30] Hosseini, M., Pettifer, S., Georganas, N. D., *Visibility-based interest management in collaborative virtual environments*, in *Proceedings of the 4th international Conference on Collaborative Virtual Environments*. 2002: Bonn, Germany. p. 143-144.

[31] Quest3D. 2009 06/11/2009]; Available from: www.quest3d.com.

[32] Blender. [Accessed :06/11/2009]; Available from: http://www.blender.org/.

[33] Games, E. [Accessed : 06/11/2009]; Available from: http://www.unrealtechnology.com/.

[34] Unity. 2009 [Accessed :04/11/2009]; Available from: http://unity3d.com/.

[35] Guitar Hero, 2009 [Accessed : 04/11/2009] Available From: http://hub.guitarhero.com/