Mapping Game Engines for Visualisation

An initial study by David Birch- db805@doc.ic.ac.uk

Contents

Motivation & Goals:

The increasing power of Building Information Modelling (BIM) is bringing benefits to the Built Environment Modelling (BEM) community. The ability to visualise architectural plans in a three dimensional walk through environment aids the shared understanding of data, something otherwise challenging to comprehend. As an example we consider the locating of CCTV cameras within a public environment. Without the ability to see the view from each camera in a 3d model it is hard to verify that each camera has been placed correctly. Equally if the user is allowed to vary camera direction and location it becomes much easier to make changes to a complex design. These and many more use cases have been facilitated by the rise in the rendering capacity of BIM models.

However the move to include an ever increasing variety and volume of information within these models is pushing the limits of the rendering engines used for visualisation. For example a model of a large public building may contain up to 5 or 6 million polygons and far more if additional data layers are included. To facilitate the continued rise of BIM we undertake a study into the applicability and scalability of using Game Engines for these visualisation tasks.

An interactive application for viewing the results of this study is available online. [Birch, D. (2009)]

Assessment Criteria

In order to evaluate a large number of possible game engines we decided upon the following assessment criteria:

- ***Scalability:*** Primarily we seek an engine which will cope adequately with the scale of today's BIM models and will provide scope for future increases in their scale and detail. Good scalability will enable additional layers of data to be included with a BIM model. Scalability refers to the ability of the engine to cope with a model too large or too detailed to be displayed in a single step. For example the ability to handle a model of a hotel with hundreds of different detailed rooms, the model is very large, but only a small portion of it is shown on the screen at one time. The primary way this is supported by the engines is by streaming small portions of the data onto the graphics card as required, thus the available graphics or system memory is no longer the problem.

- ***Visual Fidelity (Artistic):*** In order to visualise data clearly and to accurately render scenes for decision making, it is essential that the engine supports a high level of visual fidelity such that models look realistic. This will work out as a requirement for advanced lighting techniques such as High Dynamic Range Rendering which eases the problem of lighting scenes with large variation in lighting levels, for example the border between internal and external spaces. It is also important to consider the level of engineering fidelity an engine can achieve, however this is pretty much uniform across the market since the engineering fidelity is primarily a problem with the data supplied to the engine and not the engine itself.

- ***Speed of Development (Usability):*** In order to gain good return on investment and to quickly prototype new models we seek an engine which will provide a fast development times.

- ***Customizability:*** We seek an engine environment which allows substantial customisation and modification — enabling creation of novel and unique features which are often required to visualise BIM layers.

Methodology

This study is intended to take place in two distinct stages. First an initial explorative study evaluating a large number of game engines, primarily via their technical specification. The goal of this stage is to find a small number of engines to be investigated in more depth. This will comprise the second stage of the study. By taking each engine in turn and attempting to visualize a large data set we hope to gain insight into the suitability of each engine for large scale BIM as well as evaluating its scalability, speed of development and visual fidelity.

Mapping Application

To enable exploration of the large space of game engines, we decided to implement a mapping application. We took advantage of a Bubble chart to map three different axes (X, Y and the radius of the "bubbles"). This enables quick exploration of a complex multi-dimensional dataset. We allow the user to choose which of the evaluation axes to map onto the chart, giving choice of how the user explores the data set. We also allow the user to click on a bubble and bring up more detailed information on each data point, showing a brief description of the game engine, the specific axis values and a brief reason for why the game engine has been mapped to these points.

The application is implemented in the Microsoft Windows Presentation Foundation (WPF) system under C# and the .Net framework. WPF applications can be rendered as a standalone windows application or via the web browser plugin Microsoft Silverlight. We implemented both options, however due to differences in API the Silverlight version is somewhat more advanced than the standalone version.

Data format

The dataset to be visualised in the application is written as an XML document. The format of the document defines three sections, the first two of which are included in a map.xml document provide the meta data for the map. The last is contained in a data.xml file which specifies each data point.

```
<?xml version="1.0" encoding="utf-8" ?>
<Map>
  <ChartConfig>
    <ChartTitle>Map of Game
Engines</ChartTitle>
  </ChartConfig>
  <Axes>
    <Axis min="0" max="5000" step="1000" unit
="£">Cost</Axis>
    <Axis min="0" max="100" step="20" unit
="%"> Features</Axis>
  </Axes>
</Map>

<Data>
   <Datum>
      <Name>Unity3d</Name>
      <Image>unity.bmp</Image>
      <Description>Unity3d is a medium size
game engine originally developed for the Mac.
</Description>
      <URL>http://unity3d.com/unity/</URL>
      <Cost value="1250">Free for academic
use.</Cost>
      <Features value="65">All essential
features, no frills.</Features>
    </Datum>
</Data>
```

- Firstly a configuration section which defines the meta-data of the chart, such as the chart title.
- Secondly an arbitrary number of axes on which to evaluate the data points are defined. These are defined with the following attributes Name, Minimum & Maximum values, Axis Step and the Unit. Each entity mapped must provide a data point for each defined axis.
- Thirdly the list of entities mapped must be provided. Each entity must give a number of standard pieces of information such as Name, URL, a longer description and optionally an image. Then for each of the defined axes a data point must be provided along with an optional description or reasoning for the value given.

An example of the data format is shown in the text box. This information is parsed by the application and mapped directly onto the bubble chart for display to the user.

Classes of Game Engines

Looking at the wide range of available game engines a few broad categories emerge. Most obviously these categories break down via cost and often though not always correlate to the features and ease of use of the rendering engine.

At the extreme end of the spectrum we find the triple "A" rated engines which are used in multiplatform blockbuster games. They are used by games studios with hundreds of people working on a title. Capable of rendering large amounts of geometry using the latest rendering techniques they represent the cutting edge of game engines. Typically these engines will come with a large suite of fully featured editors which allow visual programming, rapid prototyping and provide a myriad of interfaces for interacting with the engine without having to write code. Of course these highly desirable features come at a price, both monetary and also in terms of the learning curve of the large suite of suite of tools. Examples of engines within this category are few (due to the cost of development). However the CryTek and Unreal engines are the biggest which are currently available. The new Id tech engine may also be considered when it is released.

At the other end of the scale lie the various open source engines such as Ogre3d, Axiom, Crystal Space and Irrlicht. These are free and offer complete access to their source code allowing complete freedom to modify the engine as required. However they each suffer from a few issues to various degrees. Firstly none of these engines come with a supported editing system (although there are many attempts to provide one) and instead will require a large amount of coding to build a prototype. Coupled with this a lack of documentation dogs one or two engines in this category. Finally and somewhat unavoidable these engines are not on the cutting edge of technology and are not developing as quickly as their commercial counterparts. However that said these engines do provide valuable functionality and most importantly complete control over the rendering pipeline which is often hidden within commercial systems.

In between these two extremes lies a huge variety of commercial engines. Each of these engines will come somewhere in the middle in each of the categories. Engines in this space tend to specialise and target a specific section of the market providing a tailored set of features often at the expense of general applicability or cutting edge rendering. For example the Quest3d suite is focused primarily on architectural visualisation and simulation whilst the Unity3d engine is focused on its Web and IPhone development capabilities. Most of these engines will provide an editing system which aids ease of use, perhaps even overtaking the triple "A" engines in usability since there are less features to learn.

The choice of which strata to choose an engine from is somewhat subjective and is dependent upon the requirements for the visualisation (which will depend on the exact application). For speed of development we would advocate either a triple "A" engine or a middle of the road engine due to the provided "What You See is What You Play" editors. However for extensibility an open source engine would come out on top (that said one or two of the commercial engines also provide a source licence).

Game Engines

We now detail each of the engines which we have considered. Along with a ranking in each of the following axes we considered:

Axes of Evaluation

The relationship of any of the follow axes can be explored interactively via the mapping application, with any of these axes being mapped to the X, Y and radius axes of the bubble chart.

Axis	min	max	step	unit	Description
Cost	0	5000	1000	£	The cost of the cheapest version of the game engine. Estimated if public pricing is not available.
Features	0	100	20	%	An estimation of the level of features this engine provides - normalized to the scale 0-100% where 100% is the fictional engine which "has it all".
Usability	0	100	20	%	A normalized estimate of the usability of the engine, high score is better. This is related to the toolkit provided by the engine.
Scalability	0	100	25	%	The degree to which the engine is likely to be scalable, primarily by supporting streaming.
Fidelity	0	100	20	%	A normalized estimate of the visual fidelity or photorealism which the engine can achieve.
Cross Platform	0	10	2	Platforms	The number of platforms that the engine can target, for example different consoles, Windows, Linux and Mac.
Web Player	0	100	25	%	Whether or not the engine has an associated web player for viewing via a web browser.
Customizability	0	100	20	%	A measure of the ability to customize parts of the rendering process. For example via scripting languages, large APIs, large editors, visual programming or coding against the source code/SDK.
Source	0	100	25	%	Whether or not the source code is available, either to be brought or as open source.

3d Game Studio

Description	URL
3d Game Studio is a game creation system which has a strong engine along with a good collection of editors. It also supports adaptive scene tree technology.	http://www.conitec.net/english/gstudio/3dgs7.php

Axis	Value	Description
Cost	900	Cheaper less powerful version available
Features (Completeness)	45	Average
Usability (Toolkit support)	50	Reasonable editor
Scalability	20	Streaming not supported
Fidelity	40	Average
Cross Platform	1	Windows only?
Web Player	25	False
Customizability	70	Lite-C scripting
Source	50	Source in license?

3DVIA Virtools

Description	URL
From Dassault Systemes. Contains editor, scripting language and C++ SDK. Includes a high-end web player and a large suite of supporting programs and applications.	http://a2.media.3ds.com/products/3dvia/3dvia-virtools/welcome/

Axis	Value	Description
Cost	3000	Unknown - high?
Features (Completeness)	55	Average
Usability (Toolkit support)	50	Editor, good compatibility with other 3ds tools.
Scalability	80	Streaming supported (aided by addon?)
Fidelity	50	Average
Cross Platform	5	Windows, Mac, Xbox, Web and Wii
Web Player	75	True - also multi-user
Customizability	70	Custom scripting with good tools
Source	20	No Source

C4 Engine

Description	URL
The C4 Engine supports Scene Graph Techniques, a good range of lighting techniques and several editors.	https://www.terathon.com/c4engine/features.php

Axis	Value	Description
Cost	20	Industrial version also availible
Features	55	Good
Usability	40	No editor
Scalability	20	Streaming not supported
Fidelity	50	fair
Cross Platform	3	Windows, MacOS, PS3
Web Player	25	False
Customizability	40	Graphical scripting only?
Source	50	Source in license?

CryTek

Description	URL
The Cry Engine is a fully featured game engine widely used in industry for high end computer games. It supports PC, Xbox and PS3 development and contains a procedural terrain and vegetation generator. Also supports Environmental Audio, a large suite of editing systems, dynamic time of day lighting. Offline rendering support is also included along with built in geometry streaming and performance analysis tools.	http://www.crytek.com/technology/cryengine-3/specifications/

Axis	Value	Description
Cost	4500	Unknown but high!
Features	90	Most everything you could ask for.
Usability	65	Huge number of features to master though with many editors.
Scalability	90	Supports streaming of geometry and textures.
Fidelity	90	Strong Realism
Cross Platform	3	Windows, XBox, PS3
Web Player	25	False

Customizability	70	Large modularity and scripting capacity.
Source	30	Unknown - probably expensive

Gamebryo

Description	URL
Gamebryo is a game creation system which has also been used for training simulators. It supports a wide variety of consoles and input devices. Contains a terrain generation system with built in dynamic level of detail.	http://www.emergent.net/en/Gamebryo-LightSpeed/Features/

Axis	Value	Description
Cost	3000	Unknown
Features	50	Engine focuses on generality and not necessarily rendering performance.
Usability	50	Large number of editors make game creation simpler.
Scalability	60	Latest version supports some texture streaming
Fidelity	70	Good
Cross Platform	4	Windows, PS3, Wii, Xbox
Web Player	25	False
Customizability	60	Lua and C++ scripting
Source	60	Source code license available

Irrlicht

Description	URL
The Irrlicht engine is an open source renderer. It provides a SDK in C++ and also in .Net. All work with the engine would require a large amount of coding. There are a few editors in various states of completion. Supports scene graph techniques.	http://irrlicht.sourceforge.net/features.html

Axis	Value	Description
Cost	0	Open Source and free
Features	50	Reasonable renderer
Usability	40	No editor, all must be coded
Scalability	45	Streaming supported via custom coding ontop of the engine
Fidelity	50	Average
Cross Platform	4	Windows, Linux, Mac, Solaris

Web Player	25	False
Customizability	50	No scripting but open source
Source	100	open source

Ogre3d

Description	URL
The Ogre engine is an open source renderer. It provides a SDK in C++. All work with the engine would require a large amount of coding. Supports material and mesh Level of Detail. Supports a hierarchical scene graph. There is also a port to C# - Axiom3d.	http://www.ogre3d.org/about/features

Axis	Value	Description
Cost	0	Free and open source
Features	40	Reasonable renderer, no editor
Usability	30	No editor, all must be coded
Scalability	40	Streaming likely supported via custom coding ontop of the engine
Fidelity	50	Average
Cross Platform	3	Windows, Linux and Mac OSX
Web Player	25	False
Customizability	50	No scripting but open source
Source	100	open source

OpenSceneGraph

Description	URL
Open Scene Graph is an open source engine which supports scene graph technology allowing larger scenes to be rendered. It also has support for a large number of data types. However it lacks a current editor and advanced lighting techniques. It supports "Continuous Level of Detail (CLOD) meshes "	http://www.openscenegraph.org/projects/osg/wiki/About/Introduction

Axis	Value	Description
Cost	0	Free and open source
Features	40	Below Average
Usability	40	No up to date editor
Scalability	50	Streaming supported via coding
Fidelity	35	Below Average

Cross Platform	3	Windows, Mac and Linux
Web Player	25	False
Customizability	50	No scripting but open source
Source	100	open source

Panda3d

Description	URL
Panda3d is an open source engine supporting coding in python and C++. Comes with its own profiling and shader generation system.	http://www.panda3d.org/features.php

Axis	Value	Description
Cost	0	Free and open source
Features	45	Below average
Usability	40	No editor
Scalability	20	Streaming not supported
Fidelity	45	Above average due to shader generation
Cross Platform	3	Windows, Mac and Linux
Web Player	25	False
Customizability	75	Python or C++ scripting
Source	100	open source

Quest3d

Description	URL
Quest3d is an engine focused on architectural visualisation. It does not include the wider game focused feature set other game engines do. It supports a wide range of input/display devices, and now supports publish to web. However it lacks several high end rendering features and is limited to 1200Mb memory footprint.	http://www.quest3d.com/index.php?id=208

Axis	Value	Description
Cost	2200	Licenses range from Euro 1250 to 10000
Features	70	Fair range of features
Usability	60	Editor and custom scripting.
Scalability	25	No Streaming, but will handle 6 million triangles.
Fidelity	70	Good
Cross Platform	2	Windows, Web (ActiveX)

Web Player	75	True
Customizability	60	Lau / C++ scripting
Source	20	No source code

Shiva3d

Description	URL
Shiva3d is a commercial game engine which very strong support cross platform and a web player. It offers an editing system and a good set of features.	http://www.stonetrip.com/3d-game-engine.html

Axis	Value	Description
Cost	900	£125 academic licence
Features	50	Fair range of features
Usability	50	Reasonable editor
Scalability	90	Streaming fully supported
Fidelity	45	Fair
Cross Platform	8	Windows, Max, Linux, Iphone, windows mobile, PSP, Wii and Xbox live
Web Player	75	True - very strong
Customizability	65	Lua Scripting
Source	20	No source?

Torque

Description	URL
The Torque engine supports a wide range of platforms and has a large userbase in industry and academia. It is aimed at the mid to low end of the market however it does provide a surprisingly good feature range, including PhysX and some advanced rendering and lighting functionality. It also supports a strong web publishing system	http://www.garagegames.com/products/torque-3d

Axis	Value	Description
Cost	600	Many addons at 20-500GBP
Features	60	Reasonable renderer
Usability	65	Comes with an editor.
Scalability	25	No information, likely supported due to web player
Fidelity	60	Fair

Cross Platform	3	Windows, Mac, Web player
Web Player	75	True
Customizability	20	No scripting but source available
Source	75	Source license = £600

Unity3d

Description	URL
Unity3d is a medium size game engine originally developed for the Mac. Primary features include support for the IPhone and a web player. Scripting language support [C#, Mono, Javascript and Python] for extensibility. Also supports the PhysX physics engine and terrain and Vegetation generation. Also supports IPhone development.	http://unity3d.com/unity/

Axis	Value	Description
Cost	1250	Free for academic use.
Features	65	All essential features, no frills.
Usability	75	C# scripting and good editor.
Scalability	90	Supports streaming of geometry and textures. (Pro version)
Fidelity	60	Fair
Cross Platform	3	Windows, Mac, Web Player
Web Player	75	True
Customizability	60	Scripting in C#, Javascript and Boo. Good documentation
Source	50	Request special license

Unreal SDK

Description	URL
The Unreal Engine is widely used in industry for computer games and has just been released a free non-commercial development kit. Comes with a wide range of editors, PhysX, java like scripting, crowd AI and many lighting techniques.	http://udk.com/features.html

Axis	Value	Description
Cost	2500	Royalty license also available.

Features	90	Most everything you could ask for.
Usability	65	Large number of features to master though with several editors.
Scalability	90	Supports Streaming of geometry and textures.
Fidelity	80	Good
Cross Platform	3	Windows, XBox, PS3
Web Player	25	False
Customizability	60	Large modularity and scripting capacity
Source	30	Unknown - probably expensive

Vision Game Engine

Description	URL
Trinigy Game Engine supports a reasonable renderer with an editor and good integration with industry standard tools. Supports SpeedTree and PhysX, scripting via Lua and advanced rendering techniques.	http://www.trinigy.net/index.php?id=36

Axis	Value	Description
Cost	1000	Unknown - non-royalty also with academic licensing
Features	55	Average
Usability	50	Editor, good compatibility with other tools.
Scalability	90	Streaming fully supported
Fidelity	45	Average
Cross Platform	4	Windows, Xbox, PS3 and Wii
Web Player	25	False
Customizability	60	Lua Scripting
Source	75	Ships with source

Results:

We now discuss the results of this mapping exercise; whilst this is best done by interactively exploring the mapping application we will discuss a few of the more interesting choices of axes. The full web application is available for exploration and we would encourage the reader to try out the mapping system.(Birch, 2009)

Cost vs. Features vs. Usability

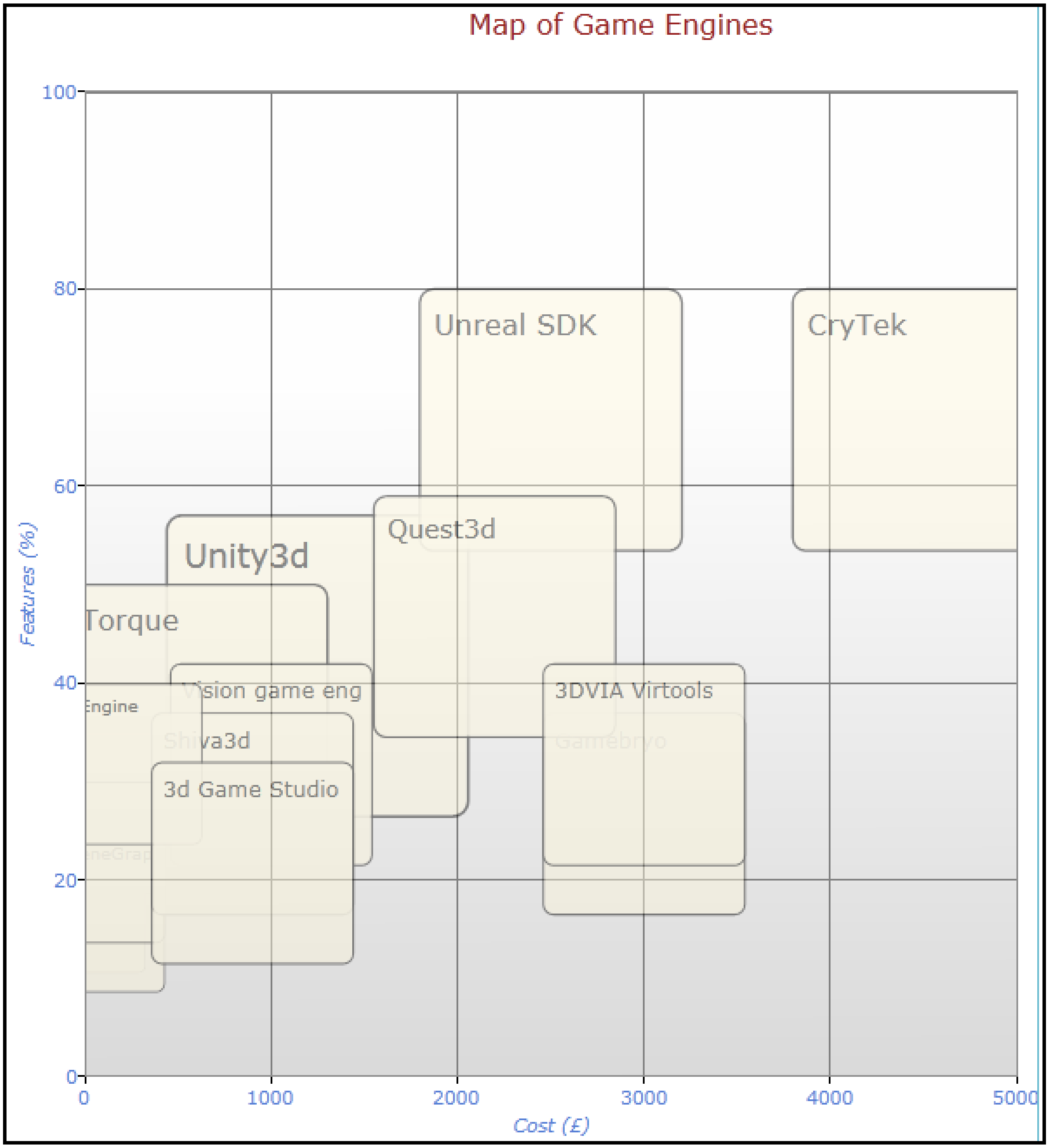
In this chart we map measures of the cost and features to the X and Y axes and map the measure of usability onto the radius of the bubbles in the chart.

Unsurprisingly the top tier of game engines (Unreal & CryTek) come out on top in terms of features and cost (which for the CryTek engine is undisclosed). They also come out fairly well in terms of usability (as shown by the larger squares) which is a result of the range of editors they provide.

After these we see a cluster of middle of the road game engines, including Unity3d, Quest3d Torque and 3DVia Virtools. These engines are somewhat hard to map since they all market themselves for different segments providing specific tools, perhaps at the expense of more general features. Thus these engines often have a large number of domain specific features which are difficult to compare across the sector.

Finally we find the open source engines which are of course free. They provide a fair proportion of the features which are common across the commercial game engines and thus do represent a viable option.

The usability metric is shown as the size of the bubble, for the open source engines this is quite small since to use these engines a large amount of code would need to be written. Alternatively we see that most of the commercial engines come with a “what you see is what you play” editing interface which makes rapid prototyping possible. With the larger game engine systems we note that this almost becomes a curse since there is such a wide variety of editors and so much functionality within them that learning the editors will take a substantial amount of time, although of course probably less than having to interface with the engine directly through code. It may be concluded that there is at least in terms of usability a sweet spot in the mid-range engines between having no editor and having a very complex series of editors.


Map of Game Engines
Unreal SDK
CryTek
Quest3d
Unity3d
Torque
Vision game eng
3DVIA Virtools
Engine
Shiva3d
3d Game Studio
Features (%)
Cost (£)
0
20
40
60
80
100
1000
2000
3000
4000
5000

Scalability vs. Fidelity vs. Features

In this chart we map estimates of scalability and visual fidelity to the X and Y axes and map the measure of features onto the radius of the bubbles in the chart.

Scalability as we use the term here is the capacity to handle large 3d models, either comprising a large volume of low detail models or a few highly detailed models – or a mix of the two. There are three ways to tackle this problem in a modern game engine.

Firstly pretty much any engine can handle a large quantity of data if running on a very powerful computer with a good graphics card and fast memory and disk sub systems. We do not consider this as a part of scalability since it has little to do with the game engine.

Secondly by efficiently implementing the game engine so that it can handle large quantities of data with minimal overhead in storage and processing power. This is of import to this discussion since any engine which is going to be scalable must be efficient. However there is a large problem in defining scalability in this manner since the efficiency of an engine is exceedingly hard to determine requiring intimate familiarity with and details of the inner workings of the game engine. One could of course perform benchmarks holding the data and machine constant to try to observe the efficiency of the game engine, however since this is a mapping exercise to decide which engines should be explored in this manner we are unable to discourse on this aspect.

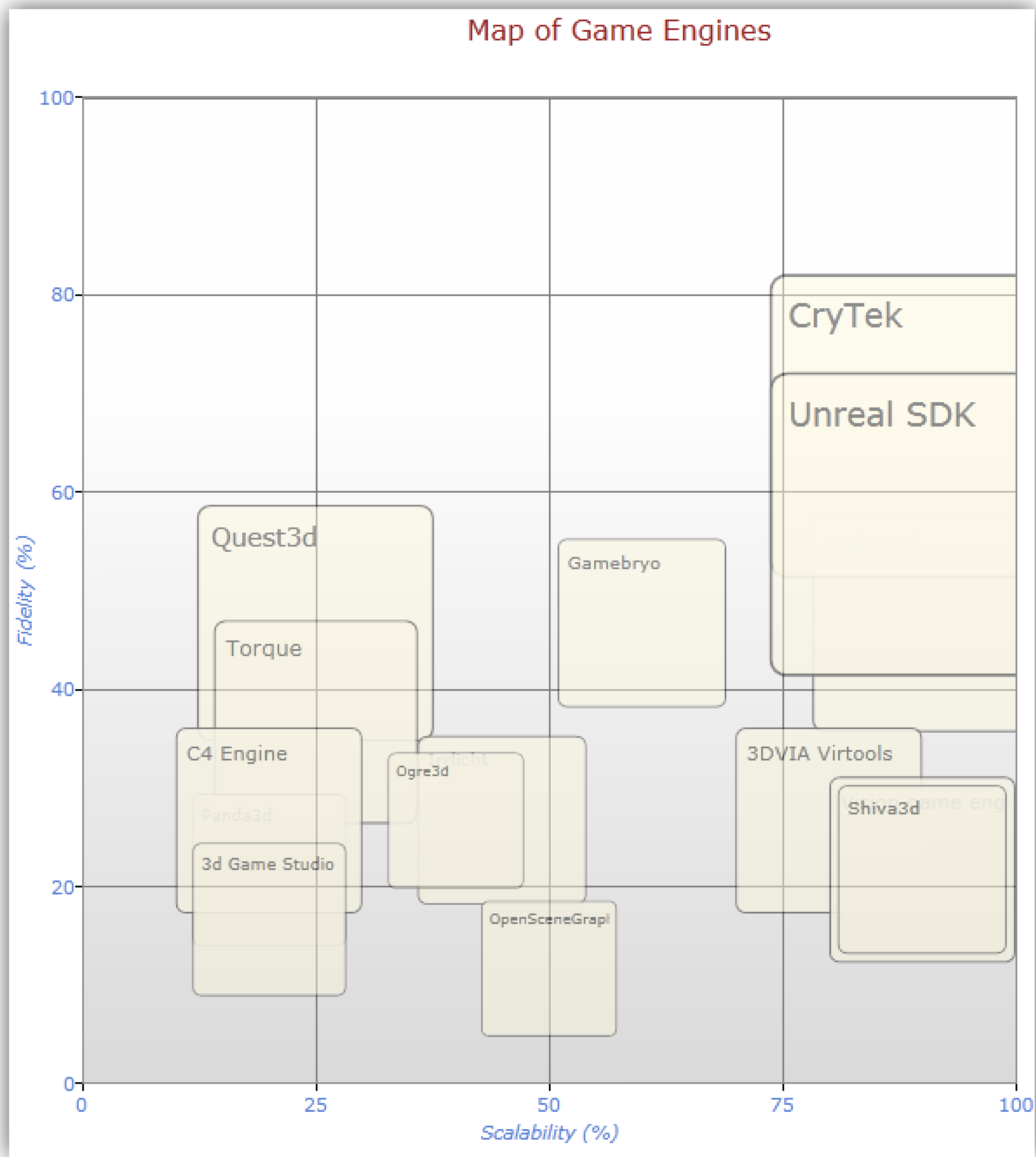
Finally and more usefully once a scene reaches a certain size it will no longer be able to fit in memory (no matter how efficient the engine is), and thus other approaches must be taken. The main approach is to use *streaming* which refers to the automatic movement of the parts of the scene being viewed in and out of the graphics card memory. Automatic streaming dramatically increases the amount of detail a scene may contain.

Within the mapping exercise we can see that the CryTec, Unity, Unreal, Shiva, Vision game engine & 3dvia tools all support some form of streaming, making them amenable to large scenes. Also we note that all of the open source engines can be made to support some form of streaming via custom programming; however it will not necessarily be automatic and thus perhaps not quite so helpful. Thirdly there are a group of engines which do not support streaming and thus will be less able to support huge scenes, although exactly where they top out will depend on how efficiently they are implemented.

We also mapped visual fidelity onto this graph. It must be noted that the measure of fidelity is rather subjective, and that any scene can be made to look good in nearly any engine providing one is willing to commit the required time and artistic talent to the process. However that said there are certain features such as particular types of lighting which will improve photorealism greatly, as will the ability of the engine to cope with large amounts of geometry and advanced shading techniques.

As expected the triple "A" do well in terms of visual fidelity due to their investment in cutting edge rendering (specifically lighting) techniques. It might be of interest to note that the CryTek engine has tended to aim for photo-realism where as the Unreal engine has tended toward fast and detailed game-play.

Many of the mid-range engines also come out well as although they lack the some of the cutting edge techniques the techniques they do use are well known and can be used to good effect. Similarly in standard usage cases the open source engines will look reasonable since they share some of the same rendering techniques. However there remains a clear hierarchy which is ascended primarily via engine cost — although hard work by the artist will help climb a rung to two whichever engine is used.



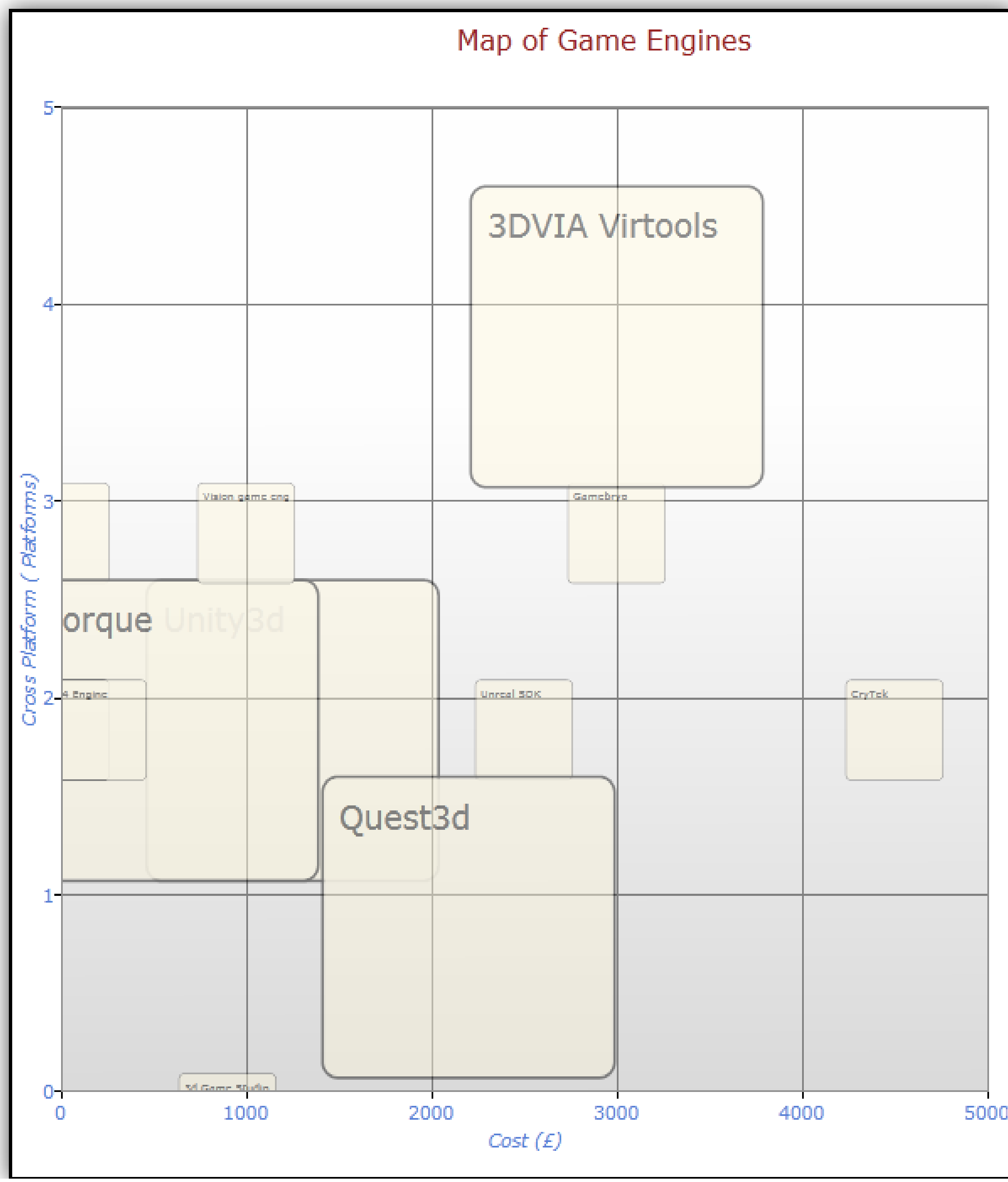
Cost vs. Cross Platform vs. Web Player

In this chart we map measures of the cost and the number of supported platforms to the X and Y axes and map whether or not the engine supports a web player onto the radius of the bubbles in the chart.

The most striking feature of this map is the size of the bubbles, a large bubble indicates the availability of a web based player for the game engine. The advantages of a web based player are three fold. Firstly that content can be located and updated centrally then accessed by the users/clients as and when required with nothing physical being distributed. Secondly the use of a web player reduces the need to ensure that a user has a specific level of hardware to render the model. However it is clear that any web player will be unable to render anything like the amount of content a specific free standing application can, nor will it be able to support the same level of visual fidelity. This may limit the usefulness of a web player to the BIM community. That said the third advantage is that it allows wide distribution of the scene allowing wide stakeholder involvement especially if using an engine such as Unity3d which supports rendering on an IPhone, which would potentially allowing access to the BIM model in real time whilst on a construction site.

The number of platforms an engine supports may also not be of wide interest to the BIM community, however it does point toward the maturity of a game engine since if it has been ported to run on different games consoles then it is likely to be well designed to be generic and stable. Of course this can work backwards as in order to provide a common set of features across all platforms the engine must find the lowest common denominator. We have counted both game consoles (Play Station 3, Xbox 360, Wii) and PC operating systems (Windows, Mac, Linux) as separate platforms. The Shiva engine is one of the most highly cross platform game engines that we have encountered, supporting 8 separate platforms. It does this by creating a platform independent API, and compressing a game and its content into a single streamable package. For each platform a separate “player” is built that can read these packaged games. This approach is quite successful and is one we expect to grow in popularity over the coming years.

Finally it is interesting to note that the number of platforms an engine supports is not necessary strongly correlated to cost. This is true since many of the open source engines support all major PC platforms (Windows, Linux and Mac) but no games consoles whilst the commercial engines mostly support just one PC platform (Windows) but also support many consoles. Since supporting a console is a costly exercise there a strong correlation between console support and cost.


Map of Game Engines
3DVIA Virtools
orque
Unity3d
Quest3d
Cross Platform (Platforms)
Cost (£)
0
1
2
3
4
5
0
1000
2000
3000
4000
5000

Source vs. Customizability vs. Usability

In this chart we map the availability of source code to the X axis, a measure of customizability to the Y Axis and a measure of usability to the radius of the bubbles in the chart.

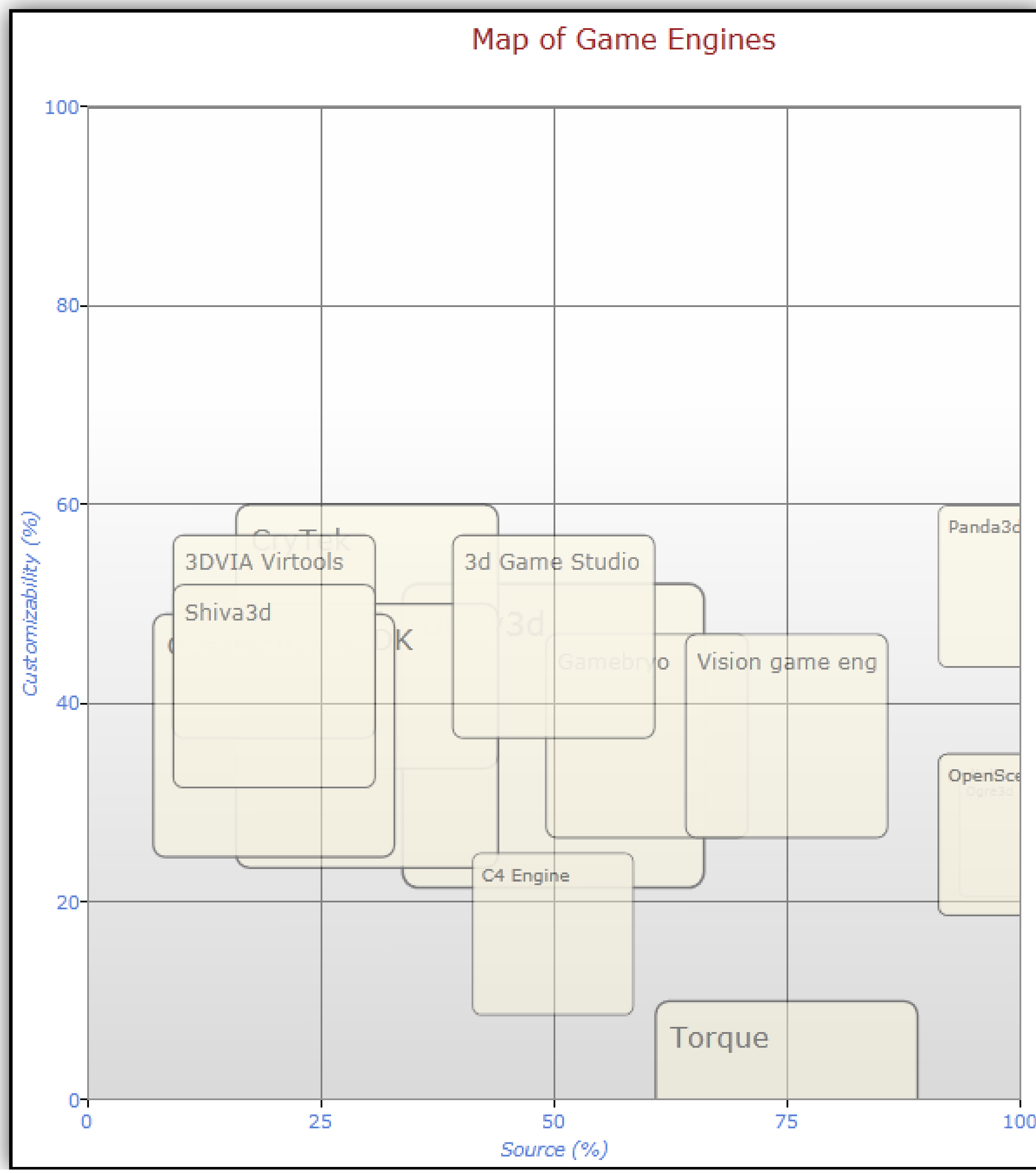
Customizability is important when constructing BIM models as there will always be some new and innovative feature which should be added to the model. For this reason it is important that the rendering engine allows this customisability. This can be done through two distinct methods. Firstly by having an open source engine which is totally extensible since any part of the engine can be modified or re-written. A second option is to provide a scripting language which allows interaction with the game engine. This higher level method is likely to provide far more rapid development, although there is an associated cost of executing the scripting over writing code within the engine. It must also be said that the power of the scripting language is strongly determined by the quality of the scripting interface provided in the SDK of the game engine as well as the documentation provided for this API. The quality of an API is hard to confirm without trying out the engine.

In the map below we see that the majority of engines offer a version which comes with sourcecode (although at up to 10x the cost). The engines which are open source do not offer any form of scripting language since it is assumed that the superior power of modifying the engine will be preferred - despite possibly higher development times.

At the other end of the scale the triple "A" engines do not mention whether they give away source code, however they provide a very fully featured tool set with visual programming which helps to automate many simpler extensions. They also provide a good set of scripting capacities for interacting with various parts of the engine. From the outside it is hard to say if this makes up for a possible lack of sourcecode which may limit the portions of the game engine which can be modified.

Between these two extremes lies by for the more common situation, commercial engines providing access to source code (at a price) but also including a scripting language. This gives the best of both worlds — the ability to carry out major modifications to the engine whilst still being able to do most jobs quickly and simply through a scripting language. Most engines support a variant of Lua, a fast embeddable scripting language. Other languages support .Net scripting or C++ interpretation all of which offer comparable productivity, though with varying levels of speed depending on how well they are implemented within the engine.

Across the board we see most engines support scripting and/or provide source code. The efficiency and power of programming with these tools is hard to quantify from the outside however a few engines stand out in this field: From the mid range commercial engines Unity3d stands out for good documentation of its .Net / Python / Javascript scripting. The 3d Game Studio also stands out for its Lite-C interpreted scripting system which should be faster than most other forms of scripting languages. Finally from the open source engines Panda3d is the only one to provide scripting support via a python interface.



Map of Game Engines
Customizability (%)
Source (%)
0
20
40
60
80
100
0
25
50
75
100
3DVIA Virtools
Shiva3d
3d Game Studio
Vision game eng
Panda3d
OpenSce
C4 Engine
Torque

Conclusions

The aim of this report was to map out the viable game engines for BIM, and to evaluate the market to find a few engines for further investigation. The mapping exercise has been reasonably successful despite a lack of solid data for parameters such as scalability. We would encourage the reader to explore the mappings interactively. (Birch, 2009)

In deciding upon which game engines to investigate in more depth it is important that we sample each strata of the market since this will show whether or not a large feature space is required and indeed whether or not it is necessary to pay a high price for a "good" visualisation tool.

From the triple "A" engines the CryTek Cry Engine 3.0 stands out for its superb photorealism capabilities. The Unreal engine is also a strong contender especially due to its lower costs despite slightly lower visual fidelity.

Within the mid-range of commercial engines choice is a challenge as many of the engines provide at least superficially similar performance levels and features lists. The Quest3d engine stands out since it is targeted toward the architectural visualisation community and thus ought to be well suited for displaying architectural data, though oddly it lacks scalability. Due to its wide use it should provide a good benchmark for comparison with the other engines. From the remaining engines Unity3d appears a good choice for the first engine to tryout due to its simple user interface, C# scripting and free academic version, it would also be interesting to investigate its web player for wider communication of models. Other engines worthy of consideration would include the 3Dvia suite of programs due to their close integration with the other 3ds CAD programs which are often used in design work.

Finally we should consider the open source engines. Open Scene Graph would be an interesting engine to investigate due to its initiative scene manage system which is built for scalability whilst supporting level of detail and culling techniques. Alternatively the Ogre3d engine would provide a more standard rendering engine for use in this investigation – it is possible that its younger brother the Axiom engine which is a port to C# of the Ogre3d engine may offer faster development time whilst providing a similar feature set.

In conclusion it is not possible at this stage to recommend a single engine for use in rendering large BIM project; however we have explored the range of possible options and narrowed down choice to a few engines which will require further investigation to provide conclusive results. This is detailed in the further work section

Further work

As mentioned in the introduction to this study this report is only the first step in the investigation. Having successfully investigated the space of available game engines we now need to investigate a good sample of the market in more depth using standardised tests to get a handle on the real life performance of each engine in the categories which we have already discussed.

To this end we intend to take a large architectural model and attempt to render it within each engine. We will document the difficulty (or hopefully ease) with which this is achieved and the level of performance and visual fidelity which can be achieved with the engine given a reasonable level of work.

As mentioned in the conclusion, time permitting we intend to investigate the following engines:

1. CryTek Cry Engine 3.0
2. Quest3d
3. Unity3d
4. Open Scene Graph
5. Unreal SDK?
6. Ogre3d or Axiom?

Bibliography

Birch, D. (2009). *Map of Game Engines*. Retrieved from http://www.doc.ic.ac.uk/~db805/GameEngines/GameEngineReport.pdf