Arts et Metiers ParisTech, IVI Master Degree, Student Research Projects, 2010, Laval, France.
RICHIR Simon, CHRISTMANN Olivier, Editors. www.masterlaval.net

Virtual Reality software taxonomy

ROLLAND Romain[1], RICHIR Simon[2]

[1] Arts et Metiers ParisTech, LAMPA, 2 Bd du Ronceray, 49000 Angers – France
[2]Arts et Metiers ParisTech, LAMPA, 2 Bd du Ronceray, 49000 Angers – France

rrolland.50@gmail.com, pi.laval@ensam.fr

Abstract— **Choosing the best 3D modeling software or real-time 3D software for our needs is more and more difficult because there is more and more software. In this study, we help to simplify the choice of that kind of software. At first, we classify the 3D software into different categories we describe. We also realize non-exhaustive software's state of the art. In order to evaluate that software, we extract evaluating criteria from many sources. In the last part, we propose several software's valuation method from various sources of information.**

Keywords: virtual, reality, software, taxonomy, benchmark

I. Introduction

The amount of 3D modeling software and virtual reality software has considerably increased these last years. Between open source software, high-end software and real time engines, it becomes difficult to estimate which software is the more suitable for a specific need.

In computer science, a benchmark means making a test to estimate and compare the performances of a system (software or hardware) with other systems.

According to [1], virtual reality is a scientific and technical domain using computer science and behavioral interfaces in order to simulate, in a virtual world, the behavior of 3D entities, which are interacting in real-time with each other and with one or several users within a pseudo-natural dumping through sensori-motor channels.

Many 3D computer graphic designers and developers test several software before beginning their work. This is actually the best solution to know if software is appropriate and if it meets user's expectations. However, realize a quality test have many downsides. First of all, it can take a lot of time if we want to test the software at most of its possibilities. It is also difficult to cover and use all the features of software, most of the users do not run the totality of tools proposed for given software. To avoid these inconveniences, we realized a comparative study of benchmark already made by professionals. This study aims at realizing a state of the art of existing software used in virtual reality and drawing up a short list among various 3D modeling and virtual reality software to make easier the choice for graphic designers and programmers. This research is not fully exhaustive but we have try to cover at best the software mainly used by 3D computer graphic designers.

At first, we are going to list the major software on the market and to group them according to various user profiles. Then, we are going to define various criteria to be able to compare them. To finish, we will present the perspectives of this study.

II. Different kind of 3D modeling software

There is a very large number of software in the field of 3D creation and animation: some have the same features and others are complementary. To find a way in this growing offer, it is possible at first to gather the software offering similar modeling and animation functions then to classify them. an in-depth study of the listed software shows two big categories:

- Tools dedicated to the creation of 3D objects, meaning the creation of these objects in a three dimensional space. This software allows applying materials and textures to these objects in order to "dress" the 3D object. Some of them also allow to create 3D animations, for example, to realize a 3D simulation (non interactive video).
- Tools dedicated to the creation of real-time 3D applications. The real-time 3D allows the users to manipulate and interact instantaneously with the 3D objects of the application. The constraints of this type of application force the creation of objects without any artifact for a maximal optimization to preserve resources of the processor at the most. Conversely to the image returned by a "classic" 3D animation, in real-time all the images are computed while the user manipulates the object or travels in the virtual

environment. We can speak about virtual reality when we simulate the behavior of objects in a virtual world. Most of real-time 3D applications are potentially virtual reality applications.

The complete list of software is presented at the end of this paper.

We will now describe more precisely the two big categories of software.

A. 3D modeling software

As mentioned before, all 3D modeling tools do not perform all the same functions, so it would be interesting to categorize these software. To realize this classification, we were inspired by several lists of 3D software on internet, and more particularly by the work of Benoît Rogez, who realized on its personal website a heuristic map of the various software related to 3D[1]. His work categorizes 3D software in 14 different categories which contain themselves several subcategories. We selected applications which seemed the most useful for the analysis and chose to classify that software according to 6 categories:

1) General modeling software

This kind of software allows creating 3D objects. 3D modeling software is based essentially on the manipulation of primitive objects. These can be in 3D (cubes, spheres, cones), but also in 2D (rectangles, Bézier curves, NURBS). The software proposes a set of tools which allows transforming primitives objects to more complex objects like buildings or characters. This kind of software can also modify other attributes such as texture of the object, its lighting or its transparency. This software does not contain tools for performing animations and are not really visible among "all in one" solutions, which contain tools for animation and other features. Among that software, we can name Hexagon made by DAZ 3D, AC3D from Inivis and Wings 3D.

2) CAD software

CAD software (Computer-Aided Design) allows to conceive complex systems, like cars or buildings and to visually and technically estimate the system, in the closest conditions to physical reality. This software includes tools to assist the engineer in the conception of a product. It is the first step to realize mechanical parts or industrial systems. We list in this category all architectural tools (architecture of buildings) such as SketchUp made by Google or Revit Architecture from Autodesk and design and engineering tools such as Solidworks made by Dassault Systèmes or Autocad from Autodesk. We also list Product Lifecycle Management software. The PLM consists of an optimal Information Technology management of all the stages of the product's life, since the expression of the need, up to the withdrawal from the market [2]. We can name software like CATIA from Dassault Systèmes, NX from Siemens and also ProENGINNER made by PTC.

3) Digital sculpting software

Digital sculpting or organic modeling software are different from classic modeling software. Indeed, they allow "sculpting" a 3D object in real time. This technique allows an intuitive and a direct modeling, as if the user works with modeling clay or wax [3]. This method of work has become more and more common to finalize or detail a 3D object, in particular for characters and creatures. This kind of software has two distinctive features: on the one hand, it is able to calculate and show a huge number of polygons at the same time without slowing down the computer. On the other hand, once the model's editing ended, the user will often produce one "normal map" applied to the low resolution 3D object. This technique will give a relief illusion similar to the high resolution 3D object, keeping a reduced number of polygons. software the most considered in this domain are Zbrush from Pixologic and Mudbox from Autodesk. There are also Silo made by Nervercenter, Argile from N-Sided, 3D Coat from 3D Brush and CB Model Pro made by Dassault Systèmes.

4) Humans and creatures authoring software

That kind of software allows creating quickly characters, animals and creatures in a different level of quality. These models are easily editable, exportable and user can change their poses thanks to dedicated tools. Then they can be exported towards other software to animate them, to integrate them into a virtual environment or also to edit them. The three mainly used solutions are Poser from SmithMicro, Quindam from N-Sided and DAZ studio made by 3D DAZ. There are also MakeHuman, an open source software from MakeHuman Team and FaceGen Modeller from Singular Inversions Inc (this last one is specialized in faces creation).

5) Landscapes and vegetation creation software

This kind of software is specialized on 3D overview. It allows creating 3D landscapes and vegetation. Artificial terrains generated by the application are created through a mix of procedural methods and user painting of height fields [4]. The procedural methods, usually based on fractal subdivision, are controlled by user specified parameters. Erosion processes can be simulated on these surfaces [5]. It is also possible to import 3D objects realized in other software. We can name Vue d'Esprit from E-on Software, 3D Bryce from DAZ 3D, WorldBuilder made by Digital Element and Terragen from Planetside.

6) "all in one" software

These software are complete 3D solutions. They cover the complete creation process of cartoon movies or 3D contents for video games, within a single software. This software contains, at least, tools for the modeling, animation, lighting, texturing, render and special effects. Some of them also allow managing physics, fabric, hair and fur materials as well as animation of fluids. Some of them can also animate 3D characters (physical and facial features). This kind of software is widely developed and used. Among the reference software, it is important to name 3D Studio Max, Maya and Softimage (XSI) from Autodesk, 3D Lightwave from NewTech, Cinema 4D from Maxon, Carrara made by DAZ 3D, Houdini from Side Effects and Blender, an open source software from Blender foundation.

[1] http://www.shadows.fr/mind-map-logiciels-3d

B. *Virtual Reality software*

According to [6], virtual reality software has to contain 3 modules, which are:

- The render, which consists in transforming information of the virtual environment into a readable image by displays.
- The interaction with the virtual world.
- The simulation, which defines the behavior and the evolution of a virtual reality application.

A virtual reality software may run infinitely, or at least until the user decides to stop it. This is called a loop. In order to produce an interaction with the user, virtual reality software contains 3 stages repeated in the loop of the program:

- Capture of the user's actions
- Processing of the action by a simulation engine and output of an adequate answer.
- Broadcast of the answer towards the user.

Those stages and the loop can also be applied to any real-time 3D application like 3D game engines and rendering engines. We choose to include 3D game engine and rendering engines in the virtual reality software class because they can be potentially used as virtual reality software.

#In [7], Michael Zyda said that Virtual Reality field is transitioning into work influenced by video games and thus now influences that industry as well. Because much of the research and development being conducted in the games community parallels the Virtual Reality community's efforts, it has the potential to affect a greater audience.

However, we can distinguish two trends in this kind of software. On one hand, we have software initially created for video game, and on the other hand, we have software created for industrial and scientific domains using virtual reality devices: #

1) *Real time 3D game engines and rendering engines*

According to [8], a 3D game engine should provide these features:

- Its outputs are graphic, music and sound effects and its input is the input device.
- It provides the algorithm to make the character move.
- It controls the various topographies and plays the role of artificial intelligence
- It supports the network and monitors a lot of things in the network.

In order to complete these roles, a 3D game engine is composed of different sub-engines. The rendering engine displays the object on the screen. The animation engine animates it in real time and expresses the object's motion. The physical engine deals with the object's gravity, weight and collision. The artificial intelligence engine controls the characters which are not controlled by users. The network engine makes each user sharing the same virtual space in order to interact each other. The 3D sound engine generates the game sound data to the game progress state. The map editor manages the 3D virtual environment like building, topography and characters [6].

Game engines 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 and training simulations [9].

So, 3D game engines can be an economic alternative to software that are really dedicated to virtual reality and provide virtual reality features. A good example of virtual reality application using a 3D game engine is CaveUT [10]. It is an Open source project which uses the Unreal Engine, which is the game engine used for Unreal Tournament (Epic Game). This project preserves the engine's built-in advantages, allows the re-use of existing game content, and allows the creation of new content using standard methods. CaveUT, which follows the principles described in the original CAVE™ system, supports a variety of immersive display strategies, from low-tech to fully stereoscopic multiscreen display.

Among the most considered game engines , we can cite the Unreal Engine 3 made by Epic Games, Unity 3D from Unity Technologies, and Torque 3D from Garage Games.

In this category, we also choose to include real time render engines. They do not allow creating complete applications but have the main role of 3D game engine. The goal is to display the information made by graphic designers to the game state in real time on the screen. However, it is possible to add extensions (like physical engine, AI engine or sound engine) which allow to create a custom 3D game engine. Among these render engines, we can cite Ogre 3D developed by Torus Knot Software, Crystal Space from Crystal Space Team and Open Scene Graph.

2) *Virtual Reality software*

Virtual Reality software can be similar to 3D game engines, however, it contains additional tools dedicated to integration and management of virtual reality devices such as head mounted devices, professional haptic devices, workbenches or CAVE© systems. Most current virtual reality environments provide 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 either through the use of standard input devices such as a keyboard and mouse, or through multimodal devices [11] as previously cited. The market of multi-interfaces virtual reality software remains rather limited, quality platforms are expensive and some research laboratories prefer to develop their own applications. Among the most known, we can cite 3D Via

Virtools from Dassault Systèmes, Vizard made by WorldVIZ, 3D Quest from Act-3D, Eon Reality Studio from Eon Reality, TechViz XL made by TechViz, some Open Source software like OpenMask from the Institut National de Recherche en Informatique et Automatique (INRIA) , and VR Juggler from the Iowa State University VRAC.

III. How to evaluate software

Realizing a software benchmark is a complex task: it is necessary to get a version and a license of the software and a benchmarking tool; besides, it asks for a rigorous testing methodology, which takes a lot of time. So we choose to make use of existing benchmarks, to aggregate their results by comparing them and to build our own graphs and mappings.

We will now choose evaluating criteria and propose some valuating method from comparison studies already made by professionals.

A. Choice of evaluating criteria

To realize our benchmark we have to establish a criteria comparison list. Naturally, each software does not have the same main features and users have not the same expectations. According to the kind of software, there will not be the same dominating criteria. However, we can find some common criteria among all this software.

T-o find objective criteria, we based our work on various software evaluation websites such as http://www.devmaster.net, which lists existing 3D engines or Benoît Saint-Moulin's article appeared on his website [12], who established a comparison table of 3D software. We also search on studies like “A mapping game engines for visualization” made by David Birch[13]. We extract two kinds of criteria, that are general and specific criteria.

1) General criteria

Those criteria can be applied to each software, not necessary 3D modeling or virtual reality software. These first criteria were elicited in [12] :

- The software’s first targeted market (engineering, video games, cinema…)
- The cost of licence/subscription and the annual subscription
- The compatibility with operating systems (Windows, Mac, Linux) and consoles
- The industry usage of the software (never or usually used in the first target market)
- The learning path to be productive, in months.
- The quality of company support for single user
- The quality of software’s documentation
- The compatibility with import/export needed by industries
- The number of base package for industries need
- The user communities and software’s popularity

In addition to this first set of criteria, we add these criteria:

- The software general stability
- The frequency of updates
- The ease of use which includes the training time and the interface’s clarity
- The software’s after-sales services, like tutorials, community reactivity (can be compared to software’s documentation and user communities)
- The necessity to had a professional training in order to use the software

2) Specific criteria

To realize a more accurate evaluation, we can also use criteria which are more specific and which can be only applied to a kind of software.

a) 3D modeling software

We find some interesting criteria for 3D modeling software in the CGSociety’s wiki [9]:

- The modeling tools features like NURBS, Polygon, Patch..
- The workable Polygon limit (in number of polygon) according to computer’s power.
- The supported file formats import/export (OBJ, 3DS, DXF, VRML…)
- The animation and simulation features (characters, physics, cloth, hair, smoke...)
- The texturing and painting features (UV unwrapping, vertex paint...)

In the Benoit Saint-Moulin’s article [12], we also find relevant criteria including:

- The quality of the rendering engine and the plugins (scanline, mental ray, Vray...).
- The quality of each package (animation tools, UV tools, modifiers, cloths…)
- The scripting tools quality

b) Virtual reality software

In the David Birch’s study [13], which is a comparative study carried out between Oct 2009 & January 2010 into the scalability of game engines for use in the Built Environment Modeling community, we extract these criteria:

- The speed of development, or usability, in order to gain good return of investment and to quickly prototype new models.
- The customizability, meaning the ability for an engine environment to allow customization and modification, enabling creation of novel and unique features required for a specific project.

- The visual fidelity, meaning that the engine supports a high level of visual fidelity such that models look realistic and the management of advanced lightning techniques.

We use the website DevMaster.net [14], which is a reference in Game and Graphics Engines Database. In the different features for each game engine, we have reused some information like:

- The number and the quality of tools and built-in editors
- The Artificial Intelligence features
- The lighting features and their quality
- The coding tools (pure coding, nodal or mixed)
- The special effects features and their quality (environment mapping, particle system, fire, smoke, weather…)

B. *Valuation method proposal*

Once criteria have been elicited, we can now propose software's evaluation. The ideal would have been to consider all the software, however, the number of software being consequent. In a worry of clarity, we choose to realize this test on a limited range of software for each family.

a) 3D modeling software

We choose to evaluate first "all in one" software. We compare the five most known software by the general public on a scale from 0 (worst mark) to 5 (best mark), according to three criteria (figure 1). These graphs were inspired by the work Benoit Saint-Moulin [12].

The industry popularity shows us that Lightwave 3D, Maya and 3D Studio Max are the most used in industry. Blender is the less popular, in spite of his qualities, probably because it is open source and, at first sight, because of its an unfriendly interface. The after-sales services contain company support and reactivity, documentations and training DVD's. The graphic shows us that Lightwave 3D has very good after-sales services especially with very complete documentations. Maya has also a complete documentation, with very good training DVD's. Cinema 4D has excellent support reactivity but its training DVD's are poor. 3D Studio Max has good documentation and training DVD's but its company support varies from poor to good. Blender is a special case because its support is the community with forums. However, Blender's training DVD's are low quality.

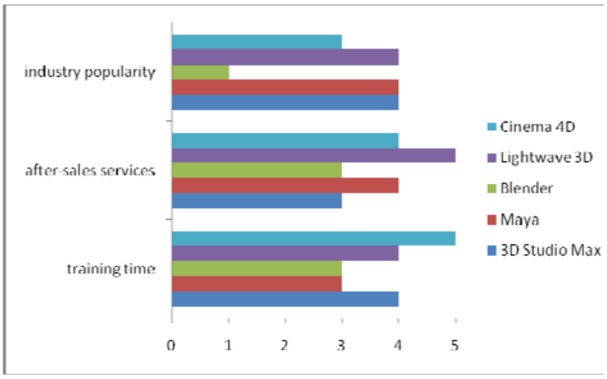

Figure 2 Comparison of "all in one" software

The training time is better for cinema 4D (less than one month) and less than three months for Maya and Blender.

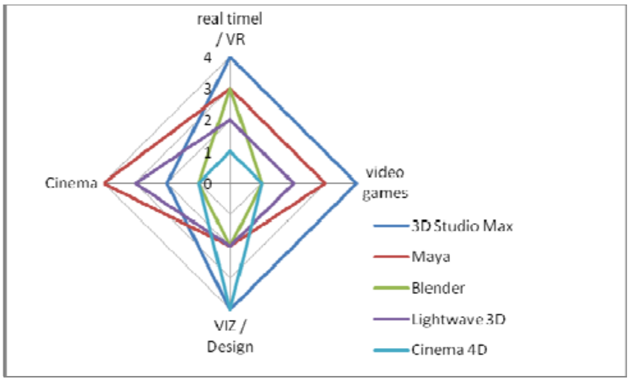

Figure 1 Popularity in various domains

We realized a second graphic (Fig.2) on the popularity of this software in four main professional domains which use these tools: virtual reality, video games, design and cinema. We can see on this graphic that each software has a main domain. Maya is better for cinema, Cinema 4D is clearly focused on design, Lightwave 3D is good for cinema. 3D Studio Max, the most well known and used, is very good for real time, video games and design; Blender is good for real time.

In the Fig. 3 mapping, we can see tools quality according to price of seven software used in 3D modeling. The tool quality of each software has been computed from evaluation of each tool (animation, UV tools, hair, cloth, Particles) with a mark from 1 to 4 [12], and converted in percentage for a better clarity. We can see in this mapping that Softimage's XSI is the best software according to his price. It is a very good software in animation, UV tools, modeling and compositing. Modo is an affordable software which doesn't have many tools but their quality

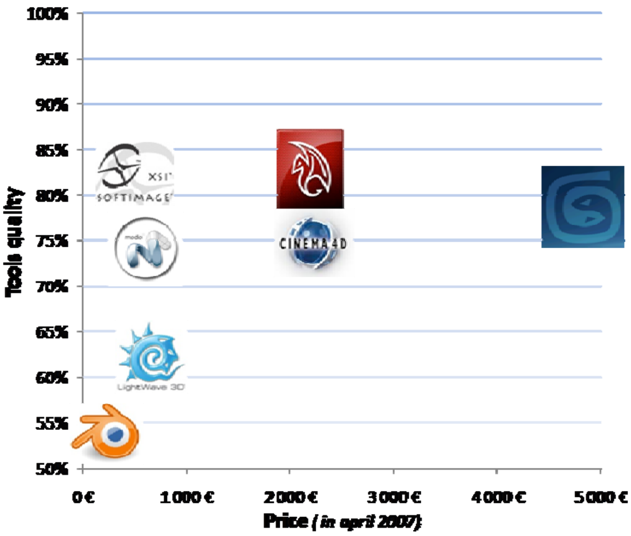

Figure 3 Prices versus tools quality

is good, especially UV, sculpting and painting tools. Lightwave 3D is also an affordable software, with good modeling tools. Blender is the best open source 3D software, most of tools are powerfull (particularly UV tool) but less intuitive than others software. Maya has the best quality tools, it is the most used software in cinema because it has first quality animation tools. Cinema 4D has very good tools in UV and painting. 3D Studio Max, the last, is the most known and also the more expensive software of the mapping, it has very good quality tools especially modeling and modifiers.

b) Virtual Reality software

To help us evaluating Virtual Reality software, we used the David Birch's work [13] and data from DevMaster.net [14].

The Fig.4 is a mapping of software's features according to the interface. Features are shown in percentage (100% mean the perfect engine which has all of features). This mapping aim helping to choose a software according to user needs and skills. To create an application with limited coding skills, it will be more convenient to choose a visual or mixed interface. For a developer, a pure coding interface will not be a problem. In this mapping, 3D Via Virtools and Quest 3D are on the side of visual interfaces, they both mainly use graphical scripting for the creation of the whole application. For small projects, it can considerably reduce development time but in big projects, it can become very complex. Concerning mixed interfaces, Unreal Engine (UDK) and CryEngine 3 are the best engines, widely used in industry for computer games. They have a large list of features and strong performances and rendering qualities. All the software, except Ogre and OpenSceneGraph are WYSIWYG editors (What You See Is What You Get) meaning 3D content displayed during editing appears very similar to the final output. Unity, Torque Engine and Shiva 3D are medium size game engines. They allow cross platform development and web publishing. Ogre and OpenScenGraph are powerfull render engines but they have, naturally, less features than game engines.

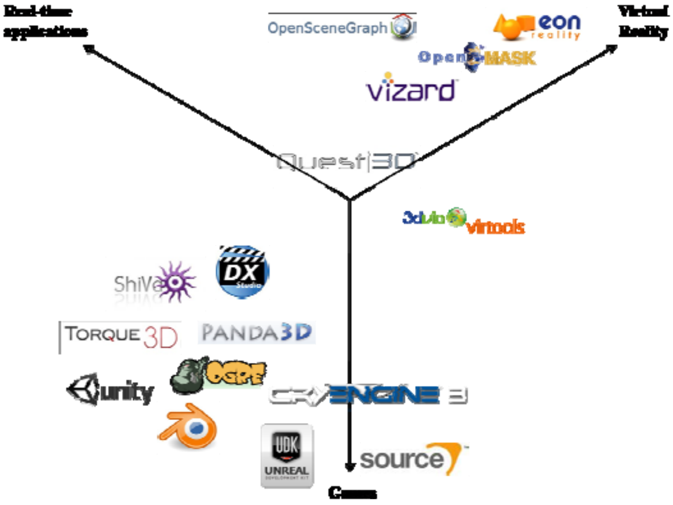

Figure 5 First targeted market

In Fig.5, we can see the first targeted market for each software. Unreal Engine, CryEngine 3 and Source engine are mainly used by video game's industry. Ogre 3D and 3D game engines like Blender Game engine, Unity 3D, Torque Game engine, Panda 3D, Shiva 3D and DX Studio are mainly used in games and however, can be used to develop real time applications. Vizard, Eon reality and the open source software Open Mask are used for immersive and non-immersive virtual reality applications. Quest 3D, initially used for architectural visualization and 3D Via Virtools, initially used for video games, can work in the three markets but they also very used to create virtual reality applications. OpenSceneGraph is an open source render engine used for real time application and virtual reality. In this mapping, we can notice that there is no software dedicated to generic real time applications, most of software are centered on video games or virtual reality, because they are the biggest markets.

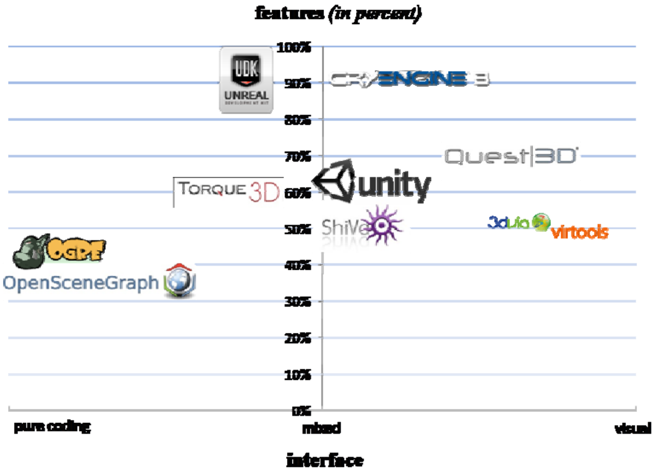

Figure 4 Features according to the interface

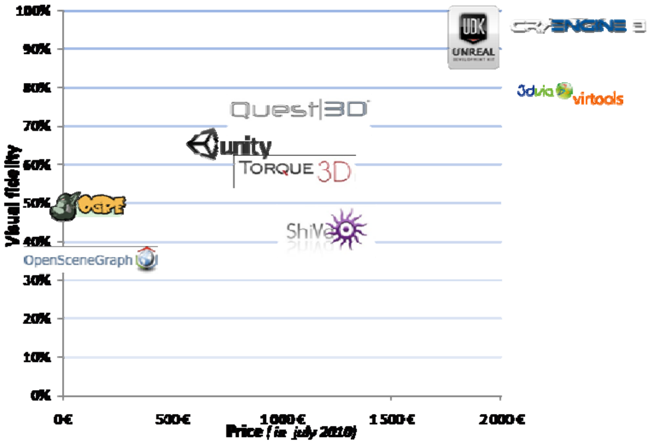

Figure 6 Visual fidelity according to price

The Fig.6 shows us a mapping of visual fidelity versus the price of software (for the less expensive professional license). The visual fidelity is, according to David Birch, a normalized estimation of the visual fidelity or photorealism which the engine can achieve [13]. In this mapping, Ogre 3D and OpenSceneGraph are free. The visual fidelity of Ogre3D is better than OpenSceneGraph because this last one has a lack of a current editor and advanced lightning techniques. Unity 3D, Torque 3D and Shiva 3D have very similar visual fidelity with advanced rendering systems, but Unity 3D seems to be the best compromise according to the price. Pro version of Unity is €880 (Free for non commercial use), Shiva editor Advanced version is €1 199 and Torque 3D individual Pro license is $1000. Unreal Engine SDK provide a non-commercial free version and a commercial version at $99 or $2 500 with a percentage of royalties over $5,000 of revenue. CryEngine has also a very good visual fidelity but the license price is unknown. 3D Via Virtools is around to €10 000, and has an average visual fidelity. Quest 3D has a good visual fidelity, especially for advanced lightning. The creative edition of Quest3D is at €1249 and the VR edition (with virtual reality devices management) at €9999.

IV. Conclusion and discussion

In this study which aims to simplify the choice of virtual reality or 3D modeling software, we realized at first a non exhaustive software's state of the art. Then, we realized a classification of the software offer. Then, we extracted a list of criteria from various sources with the aim of estimating this software.

In this paper, we have realized mappings and comparative study from various sources of information. It result from a probability of imprecision, therefore, the most important in this paper is the software evaluation methodology. We had no resources and time to do the software's tests ourselves, but it will be interesting to carry on our work and realize the tests in order to have more accurate results.

On the other hand, we chose to realize a categorization containing two kinds of software for virtual reality. It would also be interesting to add into the classification others kind of software's like collaborative virtual environments platforms. According to Wikipedia, collaborative virtual environment are used for collaboration and interaction of possibly many participants that may be spread over large distances. Typical examples are distributed simulations, 3D multiplayer games, collaborative engineering software… To create collaborative virtual environments, we can use software such as Forterra's On-Line Interactive Virtual Environment (OLIVE™), Openspace 3D from the company I-Maginer and Open Simulator. Online software can also be included in the classification. Those software can be used online as software installed on a computer. For example, http://www.avatara.com/ allows creating a high quality 3D avatar, animating it and saving it.

To conclude, we can say that the work on this paper is not set, because software are always evolving. To continue this work, it will also be interesting to create an online database, like DevMaster.net, with a list of software used for Virtual reality (beginning with our list) where anybody could give a mark to software he already used.

References

[1] P. Fuchs, G. Moreau, J.P. Papin, Le traité de la réalité virtuelle, first edition, les Presses de l'Ecole des Mines de Paris, 530 pages, 2001.

[2] Y. Gardan, M. Hilléreau and E. Perrin, CAD interactions for increasing designers awareness of manufacturability. In Parisa Ghodous, Rose Dieng-Kuntz, and Geilson Loureiro, editors, 13th ISPE International Conference on Concurrent Engineering: Research and Applications, volume 143 of Frontiers in Artificial Intelligence and Applications, 18-22 september 2006, Antibes, France, pp. 568-575, IOS Press.

[3] Tinsley A. Galyean , John F. Hughes, Sculpting: an interactive volumetric modeling technique, ACM SIGGRAPH Computer Graphics, v.25 n.4, p.267-274, July 1991.

[4] Roettger, S. and Frick, I. The terrain rendering pipeline. In Proc. of East-West Vision '02, pages 195 – 199. 2002.

[5] Brosz, J., Samavati, F. F., and Sousa, M. C. Terrain synthesis by-example. In GRAPP, 122--133. 2006.

[6] S. Jubertie, Modèles et outils pour le déploiement d'applications de Réalité Virtuelle sur des architectures distribuées hétérogènes, Thèse de Doctorat d'Université, Université d'Orléans, december, 2007.

[7] Zyda, M. From visual simulation to virtual reality to games. Computer, 38(9), 25-32. 2005.

[8] Seung Seok Noh , Sung Dea Hong , Jin Wan Park, Using a Game Engine Technique to Produce 3D Entertainment Contents, Proceedings of the 16th International Conference on Artificial Reality and Telexistence--Workshops, p.246-251, November 29-December 01, 2006.

[9] Comparison of 3D tools – CGWiki, http://wiki.cgsociety.org/index.php/Comparison_of_3d_tools, last visit on July 2010.

[10] Jeffrey Jacobson , Michael Lewis, Game Engine Virtual Reality with CaveUT, Computer, v.38 n.4, p.79-82, April 2005.

[11] Nijad Al-Najdawi, Introduction to Visualization using Game Engines, Electronic and Electrical Engineering Department, Loughborough University, Leicestershire, 2007.

[12] Benoît Saint-Moulin, TDT 3D - 3D software comparisons table, http://www.tdt3d.com/articles_viewer.php?art_id=99, nov 2007.

[13] David Birch, Mapping Game Engines for Visualisation, Department of Computing, Imperial College London, Janiary 2010.

[14] DevMaster.net – Your source for game development, http://www.devmaster.net/engines/, last visit on July 2010.

TABLE I. VIRTUAL REALITY SOFTWARE

3D modeling software	
All in one	
3D Studio Max	Autodesk
3DCanvas	Amabilis Software
Amapi	E-frontier
Amorphium	EI Technology Group
Anim8or	R. Steven Glanville
Art of illusion	Peter Eastman
Blender	Blender fondation
Carrara	DAZ 3D
Cheetah3D	Dr. Martin Wengenmayer
Cinema 4D	Maxon
EQUINOX-3D	Gabor Nagy
Houdini	Side Effects Software
Lightwave 3D	NewTek
LightWorks	Lightworks UK Ltd
Maya	Autodesk
Messiah	pmG Worldwide
Milkshape 3D	Chumbalum Soft
Modo	Luxology
moonlight3d	The Moonlight|3D team
Pixels3D	PiXELS Digital
ppModeler	Pierre Saunier
Realsoft 3D	REALSOFT GRAPHICS
Shade	E-frontier
Autodesk Softimage (XSI)	Autodesk
Strata 3D	Corastar Inc
TrueSpace	Caligari corporation
Vivaty Studio	Vivaty
General modeling	
3D MESH BLACKSMITH	TLKGAMES
AC3D	Inivis
Archipelis	Archipelis.com
Argon	Ashlar-Vellum
Bonzai 3D	AutoDesSys
FormZ	AutoDesSys
Graphite	INRIA
Groboto	Braid Art Labs
Hexagon	DAZ 3D
Metasequoia	Osamu Mizuno
Moment of Inspiration	Triple Squid
Wings 3D	
Zmodeler	Zanoza Software
CAD	
3DVIA Shape	Dassault Systemes
Alias Studio	Autodesk
AllPlan FT	Nemetschek
Arcad	Arcad systemehaus
ArchiCAD	Abvent
AutoCAD	Autodesk
CADVANCE	FIT
Cadwork	Cadwork informatique
catia	Dassault Systemes
Cobalt	Ashlar-Vellum
Concepts 3D	Csi concept
Domus.CAD	Interstudio
Intelliplus	DP Tech
Inventor	Autodesk
Microstation	Bentley Systems
NX	Siemens
ProENGINNER	PTC
Revit Architecture	Autodesk
Rhino 3d	McNeel
Shark FX	Csi concept
SketchUp	Google
SolidEdge	EDS
SolidWorks	Dassault Systemes
SpaceClaim	SpaceClaim
TurboCAD	MindScape
VariCAD	VariCAD
Vcollab Pro	Visual Collaboration Technologies
Digital sculpting	
3D Coat	3D Brush
Argile	N-Sided
CB Model Pro	Dassault Systemes
Curvy 3D	Aartform
Mudbox	Autodesk
Silo	Nevercenter
Zbrush	Pixologic
Humans and creatures	
DAZ Studio	DAZ 3D

FaceGen Modeller	Singular Inversions
MakeHuman	MHteam
Poser	Smith Micro
Quidam	N-Sided
Landscape and vegetation	
Arbaro	
Bryce	DAZ 3D
Dryad	Stanford Virtual Worlds Group
Freeworld3D	Soconne
Geocontrol	Johannes Rosenberg
Grome	Quad Software
Ivy Generator	Thomas Luft
L3DT	Aaron Torpy
Mojo World	Pandromeda
OnyxGARDEN	Onyx Computing
PlantStudio	Kurtz-Fernhout
T.ED Professional Terrain and Environment Editor	The Game Creators
Terragen	Planetside
Virtual Terrain Project	
Vue	E-On
World Builder	Digital Elemen
World Machine	Stephen Schmitt
Xfrog	Xfrog
Virtual reality software	
Virtual Reality	
AréVi	CERV
CAVELib™	Mechdyne Corporation
Croquet	
dive	
Eon Reality Studio	EON Reality
LinceoVR	Seac02
Open Mask	INRIA
OpenCobalt	Rich White
OpenSimulator	opensimulator
Solipsis	ANR-RIAM
Syzygy	University of Illinois
Virtools	Dassault Systèmes
Vizard	WorldViz
VR Juggler	
VRED	PI-VR

3D rendering engines	
artlantis	ABVENT
Crystal Space	Crystal Space Team
Horde 3D	Nicolas Schulz
Irrlicht Engine	Nikolaus Gebhardt
Ogre 3D	Torus Knot Software
Open Scene Graph	OSG Community
Wirefusion	Demicron AB
Real time 3D game engines	
3DGameStudio	Conitec Datasystems
Blender Game Engine	blender fondation
BlitzTech	Blitz game Studio
C4 Engine	Terathon Software
CryEngine	Crytek
CryEngine 2	Crytek
Demoniak 3D	Ozone3D
DX Studio	world Weaver
Esenthel Engine	Grzegorz Slazinski
id Tech 3	id Software
id Tech 4	id Software
jMonkey Engine	Mark Powell
Jupiter Extended	Monolith Productions
Leadwerks Engine 2	Leadwerks Software
NeoAxis Engine	NeoAxis Group
Nova 2009	Vertice
Panda 3D	Carnegie Mellon University
Quest 3D	Act-3D
RAGE	Rockstar Games
Reality Factory	Gekido Design Group
Shiva 3D	Stonetrip
Source Engine	Valve Software
Subdo	VB2S
Torque 3D	GarageGames
Vision Game Engine	Trinigy
TV3D SDK	Truevision3D
Unigine	Unigine Corp
Unity 3D	Unity Technologies
Unreal Engine 2	Epic
Unreal Engine 3	Epic
Visual3D.NET Game Engine	Realmware Corporation