COMPUTER GAME MODDING FOR ARCHITECTURE

Using Google Sketchup, a custom Ruby script plug-in and Unreal Technology.

RUSSELL LOWE
University of New South Wales, Faculty of the Built Environment
Russell.lowe@unsw.edu.au

Abstract. This paper will describe the design and evolution of an open source Ruby plug-in for Google SketchUp that allows geometry with UVW mapping to be exported for integration within a popular first person computer game (UT2004 and UT3 by Epic games). It will illustrate the advantages of this simplified workflow by discussing the range of complex geometries, lighting, atmospherics and interactive opportunities arrived at by students at the University of New South Wales School of Architecture. The paper will use examples of work generated using this new workflow to critically reflect on previous efforts in the use of computer games technology for architectural representation. An argument is made that while the instability that characterized computer game modding in the recent past can be productively embodied by experienced designers in architectural design projects instability itself and the complexities it engenders does not *facilitate* computer game modding by less experienced designers. The formal complexity and richness of experience presented by less experienced student's speaks to the advantages of using a simplified workflow while highlighting strengths and weaknesses in previous approaches.

Keywords. Computer, Game, Architecture, Sketchup, Ruby

1. Introduction

With the advent of second generation computer game level editors (such as Unreal Ed for UT2004 and the Hammer Editor for Half Life 2) game developers have shifted the emphasis away from building virtual world geometry within the inbuilt editors themselves. "Modders", users who make modifications to off the shelf computer games, now rely on third party modellers, such as 3dsMax, Maya, XSI and Blender. Given this new emphasis one might be surprised to discover that the workflow between the third party modellers and game editors is less than straight forward. While there are "work-arounds" for many of the problems associated with bringing custom content from third party modellers to level-editors modders find that the process generally fits in the range between awkward and unstable. (See links below “General Awkwardness and Instability”, along with Hoon and Kehoe, for a representative sample of these).

Expanding on this and briefly describing three Architecture studio graduation projects that utilised game engines will illustrate the context of instability within which architectural design work has been produced and its effects on the outcomes. The paper will then describe the design and development of a Ruby script for exporting custom geometry from Google SketchUp to use in the Unreal Editor. This plug-in facilitates a much more direct workflow between a third party modeller and a computer game editor than has been the case in the past. Finally, the paper will use examples of work generated using the SketchUp to Unreal workflow to critically reflect on previous efforts in the use of computer games technology for architectural representation. One theme that will be drawn out (through discussing a range of complex geometries, lighting, atmospherics and interactive opportunities) relates to the types of outcomes that occur when designers become comfortable collaborating with a media that is “in a constant technological flux” (Kinman, 2007).

2. Unstable media: behind the scenes of modding computer games for Architecture.

The shift in emphasis by game developers towards third party modelling applications would have solved many of the problems encountered by Uddin and Yoon in 2002. Uddin and Yoon found that the Quake game editor only accepted low polygon count objects, that compound objects were unable to be converted to “Brushes” (a type of geometry understood by the game engine) and that rendering algorithms used allow “low polygon rectilinear forms to be simulated much faster than curvilinear forms”. Unfortunately the solution of using third party modelling software without having to convert the geometry to Brushes would have been short lived. Their third party modeller of choice was “Gmax” but “as of October 6, 2005, Autodesk ... no longer offer[d] Gmax software.”

In a similar example, the author has used an SMD export plug-in (revised for 3dsMax 6, 7) and an application for streamlining the importation of custom content for the Half-Life 2 (HL2) engine called StudioCompiler. Both were written by a Modder who goes by the pseudonym “Cannonfodder”. Cannonfodder has not updated “his” SMD exporter past 3dsMax version 7 but it did function, seemingly without errors, within 3dsMax 8 and 9. By 2007 both the StudioCompiler and SMD export plug-in ceased to function as intended. In the case of the StudioCompiler another Modder, “Erix920”, made a fix by “simply edit[ing] the hex to make it work”. In a tutorial posted on the authors website (link to be advised) Cannonfodder’s StudioCompilerInstaller.v0.3a.exe is run, and then the resulting StudioCompiler.exe file is replaced by Erix920’s version. The tutorial is already very long at 47 steps and the awkwardness in simply *installing* this tool broadly characterises the entire process. The author currently uses a SMD exporter from Modder “Wunderboy” for 3dsMax 2008. After reading through the above it’s clear that the dependence, by Valve Software in this case, on what Henry Jenkins would call

"Participatory Culture" (2003) has some drawbacks. Jenkins also notes that with respect to the film industry "participatory culture is ... making demands on popular culture which the studios are not yet, and perhaps never will be, able to satisfy." Some computer game developers might be feeling the same pressure.

Valve Software's online distribution (and license verification) mechanism "Steam" is a particularly difficult medium to negotiate during the process of modding computer games for architecture. Steam provides users with an online "store" to purchase games as well as "community" and "tools" options. Problematically, Steam also updates the games you have installed automatically. In 2007, approximately one week before the author's final year Architecture graduation project students were due to present, Steam automatically updated the game directory structure to accommodate the release of "The Orange Box" game compilation. As mods of HL2 are particularly sensitive to folder locations this change prevented all of the students work from functioning. After a few hours of research a solution was found and files "migrated" appropriately. In addition The Orange Box release further complicated modding by adding a new version of the game engine and doubling portions of the file structure (this sounds relatively insignificant but for people focused on architectural design with little or no prior modding experience it can make keeping track of the source and destination folders of files during compilation very difficult). In terms of time the setback to the students work was negligible but in terms of additional stress at a particularly intense time of their year the impact was rather more significant. This is an example of where the game developer "studio" was unable to satisfy a reasonable demand of the "participatory culture"; that is, a stable folder directory structure.

In 2000 Eshaq and Karboulonis expected "emerging and affordable computer based real-time interactive technologies …. to enhance the design process through better decision-making, improved communication and collaboration, error reduction, spatial awareness, interactive design and real-time visualization." Eight years later, after having employed the instruments that "participatory culture" actually provided in several architectural design studios (with Architecture students in their first right through to the final year of their studies), a question that arises in this research is, 'what positive design outcomes have been *realized* by operating with, and within, a medium characterized by instability?'

3. Productively embodying instability: computer game modding for Architecture within a medium characterised by instability.

One typically understands instability in a negative light. This section will describe a series of projects by senior students of Architecture that have productively embodied instability within their graduation projects.

The overall design theme of the studio grows out of the famous quote by Coop Himmellblau (1984) where they sought an "architecture that bleeds, that exhausts, that whirls and even breaks. Architecture that lights up, that stings, that rips, and under stress, tears. Architecture should be cavernous, firey smooth, hard, angular, brutal, round, delicate, colourful, obscene voluptuous, dreamy, alluring, repelling, wet, dry, throbbing. Alive or dead.

Cold – then cold as a block of ice. Hot – then hot as a blazing wing." The students were required to construct an Architecture that develops after its physical constituent elements have been created; in other words, an Architecture that happens when physical elements come alive through interaction and human occupation.

In a project called "Periphery", which was concerned with the illegal occupation and consequent "vandalism" around the Central Railway Station in Sydney, 2007 graduating class student Andrew Wallace experimented with the physical capabilities enabled by the HL2 physics engine. Wallace presented a series of interconnected platforms hanging from a complex three dimensional truss that resembled a crystalline cloud formation. Each platform, suspended by three or four cables, swung precariously as the players' avatar traversed them. The platforms motion through space was a real-time physics based simulation. The only thing between the players' avatar and serious injury was a contrasting series of balustrades. The contrast of the balustrades lay in their absolute formal adherence to Australian building (and safety) standards. In other words, Wallace's project utilized the "real" physical capabilities of HL2's physics engine to test and critique notions of architectural safety around the periphery. By setting up an unstable system (the platforms) and then using "official" regulations in an attempt to make them "safe" Wallace highlighted the irony that the only thing holding everything aloft (the crystalline cloud truss) was itself unsupported. The floating cloud truss violated the same physical laws that the things that depended on it struggled so hard to test and uphold. In this way his project became a test of cultural as well as pragmatic laws. (See a video clip of the "Periphery" project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

George Barbas, another 2007 graduating class student using HL2, designed a multistory center for sports that lay directly over a series of railway tracks. The multiple railway tracks that passed directly through his scheme (both on ground level and elevated) were not isolated from the scheme itself but had direct formal, spatial and programmatic implications. The key conceptual subtext of his scheme concerned the relationship between the "Maze" and the "Labyrinth". The Labyrinth is characterized by a single pathway that, while filled with obstacles, will lead whoever follows it to the end. The Maze in contrast is full of dead-ends and can be understood as an obstacle in itself. The scheme by Barbas twisted a Maze around a Labyrinth. He used the real-time shifting of walls, floors and other architectural elements, which were made necessary by the passing of trains, to open up spaces where one could pass from Labyrinth to Maze and back again. In combination, and sometimes in conflict, with the functional program (a climbing wall is a good example of this duality) the Maze, Labyrinth, train and tracks demonstrated the full range of chaos and instability from the functionally useless to useful. (See a video clip of the "aLIFErec" project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

In 2008 the graduating class utilized the Unreal3 computer game engine by Epic Games. This move was in part to avoid some of the complications and instability described in section two but also reflects moves in practice by firms such as Texas based firm HKS. Talking about HKS's licensing of the Unreal 3 engine in 2007 Ian Kinman, president of the New York Society of

Renderers, says "it is exciting to see consumer gaming technology … so clearly adopted by one of the largest architecture firms in the U.S."

The instability associated with the Unreal3 engine doesn't so much lie in the collaboration between software (the UnrealEd imports a generic file format) as in the collaboration between documentation. Autodesk's recent division of 3ds Max into two versions suggests that strategies for modeling Architecture for "construction" and Architecture for "entertainment" are on different pathways. The official tutorials (on the Unreal Developers Network) along with many other tutorials by the mod community show the other side of the coin; that the "entertainment market" prioritizes object over space. It seems then, that an Architect who would mod computer games requires skills that are rarely used by their colleagues and develops custom content that is fundamentally different than the people who write the tutorials that they need to acquire them. It is within this context, that the acquisition of skills required to mod computer games relies on the combination and integration of multiple "partial" tutorials, that the next project was developed.

For his 2008 graduation project Vinh Nguyen put forward the hypothesis that a complex curvilinear Architecture, able to be reshaped over time, would present more of its surface area to the sun through the course of a day than a simple static extrusion of the site. To test this Nguyen would take advantage of the real-time cloth simulation possible in the new Unreal engine. Real-time cloth simulation represents one of the cutting edges of computing today (Nealen et al, 2006). Nguyen employed parts of "six tutorials; three for rigging and weighting in 3ds Max, a mini one for actorX to export the skeletal meshes, one for cloth and one for physics asset setup in Unreal Ed3. The major problem was with the cloth one [it] was lacking all images save for two. The trial/error approach I speak about often is because I was going off a tutorial that took me less than half way." (See Cloth Tutorials below). Ultimately Nguyen's experimentation proved that the simple static extrusion of the site performed remarkably well, only being out performed by the complex curvilinear cloth skin for short periods at the beginning and end of the day and in some periods in the winter months. In a way the "failure" of Nguyen's hypothesis released the Architecture from a primary concern for "efficiency" and encouraged a refocusing on the advantages presented in terms of form, space and light. (See a video clip of the "BioFuel" project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

4. Mitigating the instability surrounding game modding: developing a Ruby script plug-in for Google SketchUp to directly supply custom content for Unreal Technology.

While section three argues instability can be productively embodied by experienced designers in complex architectural design projects it's clear that instability itself and the complexities described above do not *facilitate* design engagement with real time digital techniques of architectural design and representation. To take advantage of computer gaming technology in the early stages of a students architectural design career a new workflow was required; one that mitigated the levels of software collaboration and instability that have characterised computer game modding to date. (In the first year Architecture Design Studio course where this approach was introduced there were approximately 200 students (comprising students studying towards degrees in Architecture, Architectural Computing and Engineering with Architecture). No prior programming experience was assumed or was necessary in the course. While a recent survey by the author (2008) showed that an average first year Architecture student had modified half of the computer games that they had played no prior modding experience was assumed in the course. Almost all students used their own laptop computers; with no university lab time booked for the course. In this situation a simplified workflow utilising free or very inexpensive software is an enormous advantage.

Google SketchUp, a very popular and free 3d modelling application, gives users the ability to produce geometry, apply textures and adjust UVW mapping within a single application. It is used in Architecture from conceptual design (Tahar, 2006) to building information modelling (Park, 2008). Along with many useful tools there is a facility to install plug-ins created using “Ruby”; an “open source programming language with a focus on simplicity and productivity”. This facility is especially important given that in standard form Google SketchUp is restricted to one export option (KMZ for Google Earth) which is not directly supported by computer gaming technology. (Changing the KMZ file extension to ZIP lets one explore the “new” archive and find a DAE, Collada, file which can then be imported into 3ds Max. Note: the authors very recent experiments have shown it is possible to import the DAE directly into UT3, to date the results are promising but still unsatisfactory). To utilize and further broaden the range of applications for Google SketchUp the author modified an existing ASE exporter to create a direct link between Google SketchUp and the UnrealEd computer game world building software.

The plug-in, called SU2ASE4UT (SketchUp to ASE file format for Unreal Technology) is a modification of an ASE exporter written by computer game modder “Belgabor”. Belgabor’s plug-in, called SU2ASE, was written for the purpose of exporting SketchUp models to create “custom scenery” items in the computer game “Roller Coaster Tycoon 3” (RCT3 by game developer Atari). Belgabor’s SU2ASE is not the first iteration of this particular SketchUp plug-in. Comments at the beginning of Belgabor’s script describes the list of plug-ins that have supplied code for SU2ASE. The list reads, “code used from the following SU exporters: SU2KT by Tomasz Marek, SU2POV by Didier Bur and OGRE exporter by Kojack”. (A compilation of these scripts can be found at: www.russelllowe.com/publications/caadria2009/caadria2009.htm). The exporters provide files for offline (“Kerkythea”, KT, and “Persistence of Vision Raytracer”, POV) and real-time (“Ogre” and “RCT3”) rendering engines. The exporters for offline rendering are much lengthier and include many more features than do the versions to facilitate online (game) rendering. For example Belgabor’s SU2ASE strips out many of the features Marek included for the rendering engine Kerkythea. SU2ASE also changes the file output from XML to ASCII.

The author’s contribution, following Belgabor, consisted of stripping out references to construction geometry and associated dialogue boxes (as these references are not recognised by the Unreal Engine), adjusting the scaling factor, adjusting the “warning” level for numbers of polygons to suit the Unreal Engine (from 500, recommended by modders of RTC3, to 25000 polygons) and writing a new “about” section to give advice on the plug-ins use (appropriate units setup etc). The result of the authors contribution is a very clear and direct (one step) provision of files ready for the Unreal Engine. When imported into Unreal Engine the SketchUp geometry is at the same scale, retains UVW mapping, references to smoothing between polygons, and receives the default material. Unfortunately, while vertex lighting gives good results in UT2004, in the newer UT3 vertex lighting is usually less successful. To obtain the best lighting results in UT3 the model requires an addition UV lighting channel. There is currently no way to achieve this in SketchUp (one must use 3dsMax, Maya, XSI or Blender).

5. The Advantages and Disadvantages of Utilising a Simplified Workflow for Computer Game Modding in Architecture.

This section will describe three projects by first year students of Architecture that demonstrate the range of complex geometries, lighting, atmospherics and interactive opportunities that are possible when significant levels of instability are mitigated by simplifying the computer game modding workflow.

Navigation over time is an underlying theme for the studio project which has the premise that “environments change over time. Action and interaction within an environment provide a vehicle to synthesize information and make sense of continually shifting structures.”

Initially it was thought that the SU2ASE4UT plug-in had a major disadvantage; that it would only recognise the application of one material to a piece of geometry. The ASCII exporter from 3ds Max in comparison will recognise multiple materials assigned to a single mesh. While this makes texturing a model with multiple materials in 3ds Max quite straight forward, it has two negative outcome; the resulting geometry performs poorly within the gaming engine and there is a missed opportunity for understanding the tectonics of Architecture. First year architecture student Sean Tran seized the opportunity to engage with his project in terms of an assembly of components. Following the initial breakthrough made by another student Yinggong Huang (Figure 1), Tran constructed a total of 55 separate elements to build an office for business people Donatella Versace and Steve Jobs.

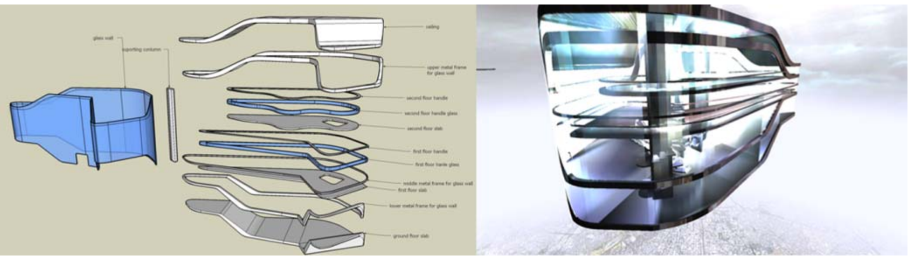

Figure 1. *Huang’s assembly of components.*

The largest of these elements used 14,088 poly’s with the office complex totalling 95,415 poly’s. The author would be the first to agree that the amount poly’s doesn’t necessarily equate to the sophistication of an

Architecture. The number of poly's *does* equate with the complexity of compound curvilinear surfaces however. Tran's design utilises a range of high, medium and low poly elements to construct a sophisticated architectural experience that curves through space in three dimensions. Further reinforcing the spatial experience Tran's elements utilise 23 different materials of which 10 were custom made using his own hand drawings. Uddin and Yoon listed the ability to construct Peter Eisenman's "House X" from "a series of rectangular and cubic 3D forms" as a "very useful feature". In complete contrast no such "useful features" exist within Tran's scheme thereby making it very difficult, if not impossible, to construct using the earlier Quake III game engine. Figure 2. (See additional images of Tran's project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

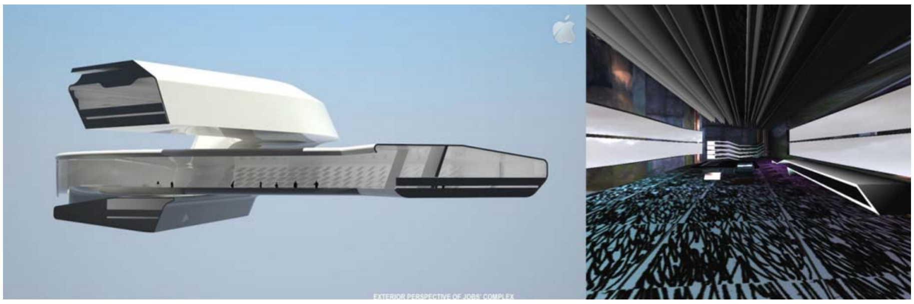

Figure 2. *Tran's interlocking spaces and custom materials.*

In the same course Cissy Miao Kang's scheme uses an aggregation of deformed spheres to create a tubular transition space. Two forces seem to be at work; an attractive force, pulling the spheres towards an imaginary centre and an opposing force that pushes outwards from the interior in an attempt to create a broadly rectilinear, and therefore conventional, space. "Natural" and "artificial" light seem to be competing in the same way. The balance between the subtlety and tension in this space is expertly handled. Uddin and Yoon would remind us that Kang's "donut" like geometry is an excellent example of what to avoid if attempting to create a Brush object (which must be convex shaped primitives). It is an enormous advantage of the Unreal game engine that it can handle collision between occupants and imported world geometry on a per-polygon basis and no conversion to Brush is required. Figure 3. (See additional images of Kang's project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

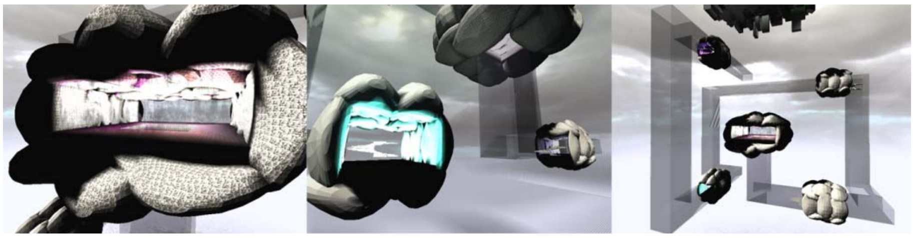

Figure 3. *Kang's non-convex geometry.*

In 2000, talking in "Modelling for Virtual Reality in Architecture", Jonas af Klercker said "there should be no need for a 'filtering' VR-expert between the architect's intentions and the layman's experiences". That Klercker's students, who were practicing Architects, used a “special VR tool” indicates there was still a gap between his expectations and reality. In 2008 first year Architecture student Vincent Hao Hsiu Hsu created a joint office complex for business people Steve Jobs and Zhang Yin. In one part of the scheme there is a seemingly flat surface, constructed from concentric rings of rectilinear elements (much like a slice through a tree trunk). As the occupant approaches the rings start to shift, triggered by one’s presence. They move upwards more at the perimeter than the center forming a subtlety curving three dimensional ramp to the next level. Other triggers make the “ramp” arc in the opposite direction. Both destinations seem unavailable to the occupant until this transformation over time. The richness of interactivity in Hsu’s scheme suggests that Klercker’s gap has closed. For now. (See a video clip of Hsu’s project at: www.russelllowe.com/publications/caadria2009/caadria2009.htm).

6. Conclusion.

The clear advantage of simplifying the workflow from third party modelers to computer game editors is the ability to facilitate the involvement of a greater range of Architects. Every few years or so a new, more sophisticated, generation of computer games emerge. The complexity of fully engaging with these opportunities rises alongside the “laymen’s experiences” and expectations. If Architects do not want a “filtering VR-Expert” between them and their clients (or them and an ability to productively embody instability in architectural design) they need to rise to the challenge. The current generation of Architecture students is doing just that.

Acknowledgments.

The author would like to extend a special thanks to the Head of the Architectural Computing discipline at UNSW, Stephen Peter, for his insight and support regarding Ruby scripting.

References.

Autodesk: http://usa.autodesk.com/adsk/servlet/index?id=5562445&siteID=123112 recovered 28 November, 2008.

Cannonfodder: http://www.chaosincarnate.net/cannonfodder/cftools.htm SMD export plug-in, recovered December 16, 2008.

Cloth Tutorials:

http://www.treadster.com/PhysX_Tut/Cloth.htm

http://udn.epicgames.com/Two/ActorX.html recovered 10 December 2008.

Character Design and Integration with HL2 and 3ds Max, NoesisInteractive.com DVD.

Unreal3 Special Edition, Bonus DVD, especially the tutorial “Physics asset setup”.

Collada: http://www.khronos.org/collada/ recovered 11 December 2008

Coop Himmelblau: 1984 Architecture is now : projects, (un)buildings, actions, statements, sketches, commentaries, 1968-1983 / [translation into English: Joe Steinbauer, Roswitha Prix]. London : Thames & Hudson.

Design Guidelines for Hospitals and Day Procedure Centres: In Australia. http://www.healthdesign.com.au/vic.dghdp/standard_components_index.htm recovered 17 February 2009.

CounterStrike: http://en.wikipedia.org/wiki/Counter-Strike

Erix920: http://developer.valvesoftware.com/wiki/Talk:Mdldecompiler recovered 28 November, 2008.

General Awkwardness and Instability:

http://forums.insmod.net/index.php?s=a07f4a96618d282876ca78358ec4068d&showtopic=16258&pid=255719&st=0&#entry255719 installing Valve SDK plug-in for Maya

http://www.hl2world.com/bbs/xsi-animation-smd-export-problem-vt30915.html XSI animation .smd problem

http://www.icecreamyou.com/blender.html#exporting Exporting from Blender to UnrealEd, note *Animations* section. recovered 11 December 2008.

Hoon, M. and Kehoe, M.: 2003, “Enhancing Architectural Communication with Gaming Engines” ACADIA22, Connecting Crossroads of Digital Discourse. pg 350.

Jenkins, H.: 2003, "Quentin Tarantino's Star Wars: Digital Cinema, Media Convergence and Participatory Culture," in David Thorburn and Henry Jenkins (eds.) *Rethinking Media Change* (Cambridge: MIT Press, 2003). Available online here: http://web.mit.edu/cms/People/henry3/starwars.html recovered 25 November 2008.

Kinman, I.: “Unreal Architecture”, Business Week, December 21, 2007 (Innovation section). http://www.businessweek.com/innovate/content/dec2007/id20071221_129534_page_2.htm recovered 11 December, 2008.

Klercker, af J.: 2000, “Modelling for Virtual Reality in Architecture.” Promise and Reality: State of the Art versus State of Practice in Computing for the Design and Planning Process [18th eCAADe Conference Proceedings / ISBN 0-9523687-6-5] Weimar (Germany) 22-24 June 2000, pp. 209-213

Lowe, R.: 2008, Beyond the boundary object: sketches, computer games and blogs facilitating design development. CAADRIA 2008 [Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia] Chiang Mai (Thailand) 9-12 April 2008, pp. 544-551

Lowe, R.: 2008, http://www.russelllowe.com/arch1101/experiment3/brief/brief_exp3.htm recovered 17 February 2009.

Nealen, A., Muller, M., Keiser, R., Boxerman, E., Carlson, M.: 2006, “Physically Based Deformable Models in Computer Graphics.” *In Computer graphics Forum*, Volume 25 Issue 4, The Eurographics Association and Blackwell Publishing Ltd. pp 809-836

Park, Hyoung-June.: 2008, Evolution + Bim: The Utilization of Building Information Modelling at an Early Design Stage, CAADRIA 2008 [Proceedings of the 13th International Conference on Computer Aided Architectural Design Research in Asia] Chiang Mai (Thailand) 9-12 April 2008, pp. 552-559

Ruby: http://www.ruby-lang.org/en/ recovered 11 June 11, 2008

Tahar, Kouider.: 2006, “Evolution or Revolution: is digital conceptual design the way forward for architects?” Computing in Architecture / Re-Thinking the Discourse: The Second International Conference of the Arab Society for Computer Aided Architectural Design (ASCAAD 2006), 25-27 April 2006, Sharjah, United Arab Emirates http://cumincad.scix.net/cgi-bin/works/Show?ascaad2006_paper4

Udin S. and Yoon S.: 2002, “Peter Eisenman’s House X, Scheme G: 3D Game Engine for Portable Virtual Representation of Architecture.” eCAADe 20 [design e-ducation] Modeling Real and Virtual Worlds Session 14.

Wunderboy: SMD export plug-in for 3dsMax 2008 – 2009. http://www.wunderboy.org/3dapps.php