

© Copyright 2015

Lily Lin Lao

Open Tours

Lily Lin Lao

A thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science in Architecture

University of Washington

2015

Thesis Committee:

Professor Brian R. Johnson, Chair

Professor Mehlika Inanici

Program Authorized to Offer Degree:

Architecture
Design Computing

University of Washington

Abstract

Open Tours

Lily Lin Lao

Chair of the Thesis Committee:
Brian R. Johnson, Associate Professor
Department of Architecture

This thesis describes the development of a 3d web based navigation tool for architecture which combines with a set of camera, animation, and rendering guidelines to facilitate and deliver a more visually consistent spaces and produce higher quality viewing experience for its users. The tool seeks to be a medium between the current products available today which either do not accept user input or provide too many options. The recommended guidelines can assist a designer in selection choices during the complicated conversion of 3d models into renderings for walk through animations, while the web based tool proposes a simplified method to fuse a combination of information for users of architectural spaces in one location.

TABLE OF CONTENTS

List of Figures	3
Chapter 1. Introduction	5
Chapter 2. Prior Work	7
2.1 3d model Components.....	7
2.1.1 Camera	7
2.1.2 Animation.....	11
2.1.3 Rendering.....	13
2.2 Technologies - tools of delivery to the audience.....	16
2.2.1 3D Authoring Tools	16
2.2.2 Adobe 3d PDF.....	17
2.2.3 QuickTime VR.....	18
2.3 Context visualization	18
2.3.1 Immersive environments – Game Engine interfaces.....	18
2.3.2 Information navigation	19
2.4 Conclusion and Observations	20
Chapter 3. Open tours Prototype	22
3.1 Concept and Key features	22
3.2 Development Tools	23
3.2.1 Software	23
3.2.2 Hardware	23
3.2.3 Scripts and Languages.....	24
3.3 Process	25
3.3.1 Modeling Development and Guidelines	26
3.3.2 Recommended Guidelines and Tips	41

3.3.3	Interface design Elements	43
3.4	Final Product and Functionality	46
3.4.1	Overview of HTML page.....	46
3.4.2	Node information.....	48
3.5	Limitations / Complications Faced.....	54
3.5.1	Time and Computing Power	55
3.5.2	Materials and Rendering.....	55
3.6	Future Possibilities	57
3.6.1	Automation	57
3.6.2	Other XML applications	57
3.6.3	Retention of Information	58
Chapter 4.	Conclusion	59
4.1	Conclusion.....	59
References	60
Bibliography	62

LIST OF FIGURES

Figure 1: Sample of rule of thirds [3]	8
Figure 2: Angle of view illustration for the human eye [3]	9
Figure 3: Camera focal length variation [4]	9
Figure 4: Visual balance example [5]	10
Figure 5: Depth of field sample [6]	11
Figure 6: Walking movement in an animation [8]	12
Figure 7: Renderer comparisons	15
Figure 8: DVIN interface [24]	20
Figure 9: Block diagram of system: front end - orange, back end - blue.....	24
Figure 10: 3d model development steps	26
Figure 11 : Rhino model, roof in yellow highlights	27
Figure 12: Maya model, roof objects are highlighted on the left.....	28
Figure 13: Camera height comparisons	29
Figure 14: Focal length tests	30
Figure 15: Library settings.....	31
Figure 16: Closet settings.....	32
Figure 17: Rules of third comparison.....	34
Figure 18: Smooth flowing animation graph appearance	36
Figure 19: Incorrect graph appearance	36
Figure 20: Light cache illustration [28]	38
Figure 21: Irradiance Map [28]	39
Figure 22: GI tab in Maya, Light Cache (left) and Irradiance (right) settings	40
Figure 23: Initial prototype design (version 001)	43
Figure 24: Kitchen XML (version 001)	44
Figure 25: XML layout (version 001)	45
Figure 26: Welcome Screen	46
Figure 27: Main Screen viewers see when starting the tour	47

Figure 28: Reconfigured node information for XML.....	48
Figure 29: XML data for bathroom node	49
Figure 30: Home screen	50
Figure 31: Lighting screen	51
Figure 32: 360 degree rotation screen	52
Figure 33: Images Scree	53
Figure 34: Map page	54
Figure 35: Material settings	56

Chapter 1. INTRODUCTION

The most common methods of architectural visualization consist of plans, drawings, and pre-rendered images and animation. Sometimes more advanced forms of visualization are produced and delivered in the form of virtual tours or fully immersive 3d environments, which provide controls for the user to be able to navigate the space. These visualization faces a number of issues due to the differences between the 3d spatial awareness in reality and computer generated visuals.

People perceive spaces in real life with a multitude of senses. Sight, along with touch, smell, and hearing help to create these imageries. These perceptions happen very naturally and are a part of our subconscious. Our eyes and mind have the ability to bridge gaps and send adjusted visual information seamlessly to the brain which automatically compensates for variable brightness and the colors in images. When a person is unable to experience a space physically, however and the visualization of architecture happens within a virtual 3d environment on a computer screen, most of these senses are obstructed and the reliance of sight alone becomes necessary. This not only changes the perception of distance and lighting for a space, the awareness and anticipation of movements and turns are no longer within a viewer's control. As a result, the timing and fluidly of movement needs to be conveyed seamlessly or the finished product lacks polish and fluid movements. This is why the knowledge and skills of developing and delivering visually appealing and consistent images and animations is so important to master.

The first issue this thesis addresses is the complexities of today's advanced software and the designer's skillset that is necessary in order to use these programs efficiently and effectively. Despite how the technical capabilities for creating architectural models have advanced, the complexities and required understanding necessary to utilize these software have also increased and multiplied. The options and menus within these programs are layered and intensive and sometimes not linked directly to any significant meaning outside of the application. Since the degree of knowledge required to attain high quality animations and renderings has increased significantly, the result is a time consuming process, often requiring a steep increase in a user's knowledge. These software also require a combined understanding of how each aspect works in

conjunction with the animation process. If the designer is lacking any of those skills, the finished product is likely to consist of low quality images that cloud and mask the intended message of an architectural piece. Although the designer has certain obligations towards acquiring fundamental skills for production and rendering, the affordance of a group of defined and structured guidelines can only be beneficial and ease their decision making process.

The second issue explored in this thesis concerns navigation and the tools the end user has available to view a 3D environment. The existing tools available offer navigational elements which are difficult to manipulate and control. Although some of the issues are results from the user's skillset and understanding of the system, some of the other causes are in the development of tools which are complicated and difficult to navigate. Within more complex systems, the manipulations of space leave a less technologically savvy user confused and frustrated with poorly selected angles of view and orientation.

This thesis proposes that the development of a new simplified web based navigational tool for touring 3d environments will bridge the gap of knowledge and counter some of the shortcomings of the interfaces available today. The production of this tool will allow the designer to more closely author the content of the design by providing assistance with a set of guidelines that are tested to produce more visually consistent and realistic camera angles and movements within a space. These guidelines would be one solution to rectify common mistakes made with camera and animation choices, such as camera angle and height as well as acceptable speed for animation. The tool will also be a space which hold other pertinent information such as lighting comparison, 360 views, and additional images and information the designer wishes to present to the user. By simplifying the controls presented to the end user, providing web accessibility, and placing multiple elements of informative data within one location, the Open Tours tool aims to provide a pleasant and more informative viewing platform while reducing stress of navigation. As a result, the final product will translate a closer vision of the designer to the user.

Chapter 2. PRIOR WORK

In order to know what elements and aspects should be incorporated into the proposed tool in order to accomplish these goals, research on cameras, animation, rendering, and past and present technologies and context delivery systems was undertaken. This research became the basis of the set of guidelines which was used during the prototype to illustrate how the application of certain steps during production can help enhance a piece. These guidelines aim to reduce the decision process for an inexperienced animation designer with the usage of cameras, animations, and renderings so that more time and effort can be applied to tailoring the goal of their narrative within their space. The first section of this chapter includes research on cameras, animations, and rendering techniques that have become important aspects of the project. The latter is a review on the technical applications and systems that informed the decision and development process of the interface.

2.1 3D MODEL COMPONENTS

2.1.1 *Camera*

Understanding when to apply the correct situation specific camera components as well as how the interactions between cameras and the human eye process information is one step toward production of good animation. Thematic elements within cinematography and photography were studied to obtain pertinent information in these topics which can help inform the choices and selections made in virtual cameras. Although the translation between real and virtual photography is not a seamless one, there are general aspects one can draw upon and learn from.

Cinematography is defined as the art of film making. Film makers study cinematography to learn how to tell compelling stories with the usage of cameras and how to weave a narrative that will engaging the audience and draw them into the story. They learn the principles of framing and rules of cinematic design which assist in their film making techniques. A finished story from a good cinematographer is able to convey a sense of richness in their environment and include compelling imagery which blend together seamlessly. Although architecture animations also use

camera, lights, and props to convey space and movement, they rarely are able to achieve the sophistication and polish of cinematographic works. Instead architecture animations frequently consist of awkwardly selected angles of view and fast motions that are visually distracting and jarring to most viewers. By understanding the core principles and rules of cinematography while also learning the beneficial application of these within architectural spaces, production of work that consist of more fluid movements which evoke the intended message of the architect and creates an emotional bond with the audience can be possible.

2.1.1.1 Rule of Thirds

One primary compositional concept common in cinematography and photography is the rule of thirds, which is used to maintain good spacing and proportion of images. The rule of thirds is defined as placement of the main subject on any of the four intersections, or power corners, made by two equally spaced horizontal or vertical lines [1] [2] [3]. This focuses the visual cues of the eye towards specific objects of interest. In the image below note how the window spans roughly the middle third of the image vertically, while the window and the building edge divide the frame into thirds horizontally, as highlighted on the right. This results in a much more visually interesting image in comparison to an image where the window is centered in the frame.



Figure 1: Sample of rule of thirds [3]

2.1.1.2 Angle of view

The average angle of view available to human eyes is approximately 130 degrees for movement but only about 40-60 degrees is directed to object recognition [3]. This corresponds closely the normal focal length of a 50 mm lens on full framed camera. As the eyes extend further outside

of this range, images become more and more distorted. It is suggested therefore that cameras should stay within a camera view of 35mm to 50 mm to mimic how people would currently see the world unless there is a unique quality or storytelling aspect which requires another specific camera angle [2] [4]. In addition, one should remember that usage of wider angles creates a larger space visually while narrower telephoto lens makes the space seem smaller.

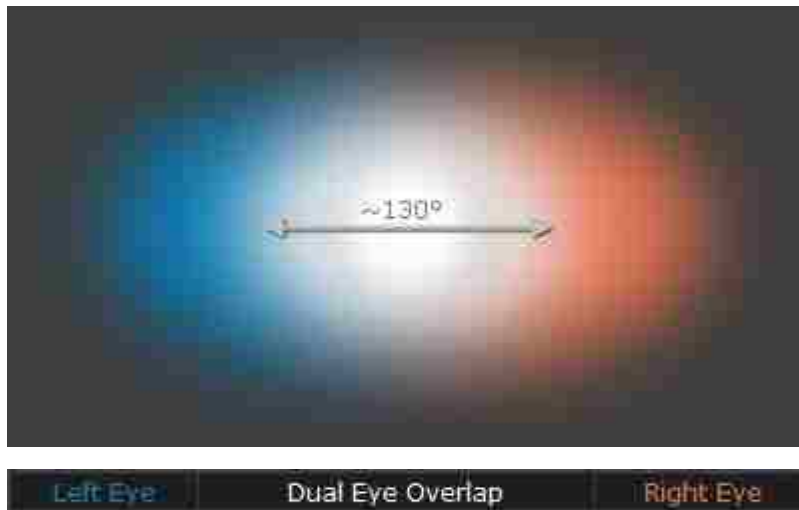


Figure 2: Angle of view illustration for the human eye [3]



Figure 3: Camera focal length variation [4]

2.1.1.3 Visual balance

Visual balance is the idea that content should be distributed within a frame so that an equilibrium state is achieved [1]. The two main forms of this balance are symmetry and asymmetry. Symmetry is easiest to achieve, one example of this is the reflection of an architectural building on the water below it. Asymmetry is a sense of balance where both sides are not identical, but the overall composition lends to a balanced frame.

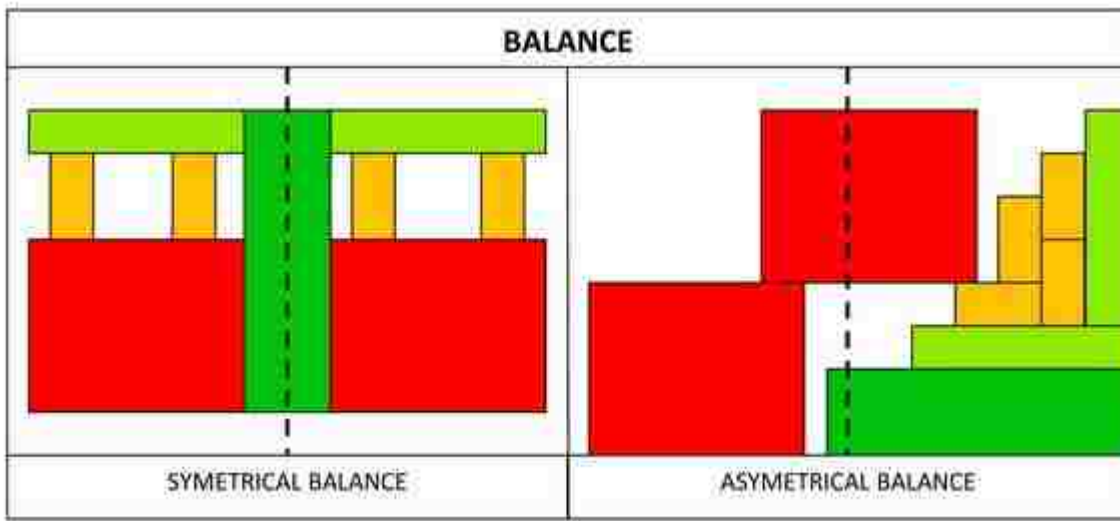


Figure 4: Visual balance example [5]

2.1.1.4 Depth of Field

The image in an idealized pinhole camera is perfectly focused everywhere, but real cameras use a wider aperture to gather more light, requiring lens optics to focus the light. This focus is only perfect at one distance. Depth of field is defined as the distance which determines the clarity or focus of an object, with the objects in the foreground and background being more blurry. Although it is an important camera feature that is often overlooked, this technique is extremely effective by providing an ability to draw the eyes to what needs to be attended to at a given moment in time. It has the ability to greatly enhance the storytelling aspect of an image since it can draw the viewer from one element to another. One beautifully developed film created by Alex Roman shows an intricate use of depth of field to focus the eyes on the elements given a specific moment in time. The movements of the animation coupled with the focus of the camera

are smooth and elegant in its transitions, one example that illustrates how the usage of these techniques can be powerful in translating the thoughts of the designer if used correctly.

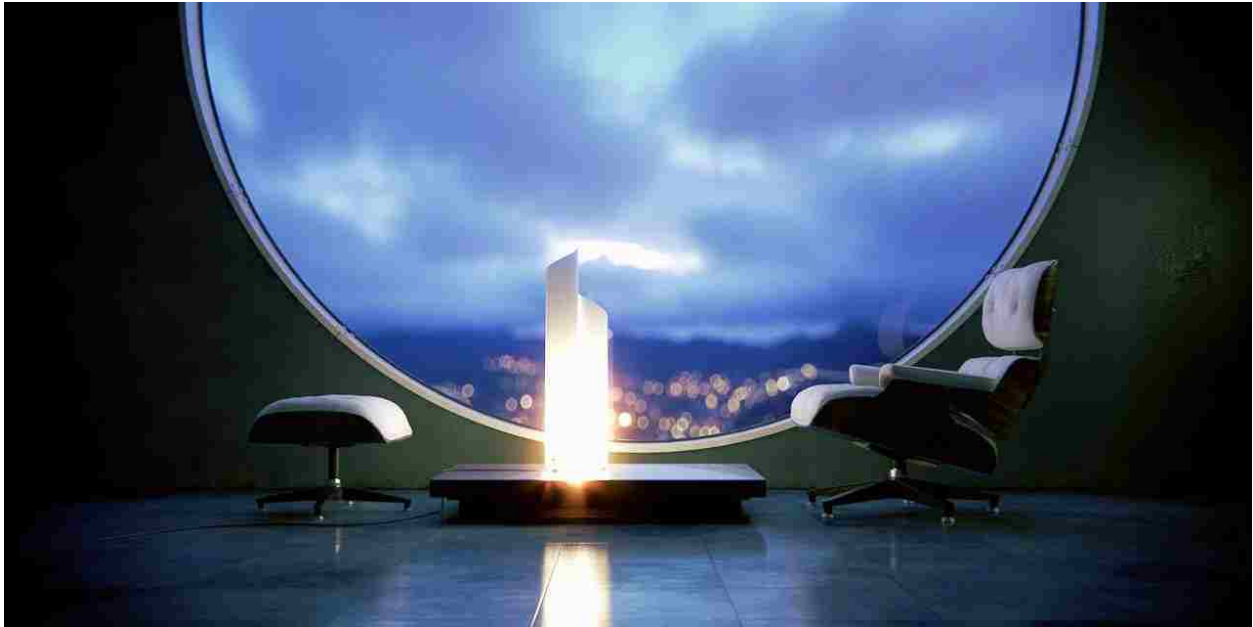


Figure 5: Depth of field sample [6]

2.1.2 *Animation*

Animation is the illusion of movement generated by a series of images played in order. Within architecture, cameras are the primary source of movement since the focus of animations in the field are views of spaces and building. Several of the 12 principles of animation developed by Frank Thomas and Ollie Johnston, [7], are quite relevant within the context of architectural animations. This research focuses on the animation principles which most directly influence architecture: anticipation, staging, slow in and out, arc, and timing.

Anticipation is a concept within animation that is often missed within architecture since the primary objects which generate movement are cameras. This thesis argues that this is the exact reason why it is important within architecture. When analyzing the process which occurs when a person turns their head in real life to view something, the anticipation or knowledge of the turn is known prior to the actual movement. In fact, the eyes look before the head turns, integrating another layer of anticipation in the action. In comparison though, when a virtual

cameras turns, knowledge that it will occur is not communicated to the viewer. This results in missed instances of awareness while the mind adjusts to take in the information.

Staging is a principle that is similar to the concept within cinematography and theatre, used to draw a viewer's attention on what the designers chooses. This technique is important in architecture, especially in cases where the designer wishes to highlight important components of an architectural piece. In addition, the smart use of depth of field combined with lighting and textures can create the intended moods, atmosphere, and story within a scene.

The principles of slow in and out, arc movements, and timing are all important components to making believable animation. This is because these movements mimic the characteristics of people and animals in reality. The appearance of animation graphs for these movements should look more like a flowing curve rather than consisting of sharp and abrupt angles. See the image below. It shows the movement of a walking animation reflected in a sinusoidal curve, the slower steps in the beginning and the end of the pattern, and the movements in the middle that maintain a steady and smooth up and down pace.

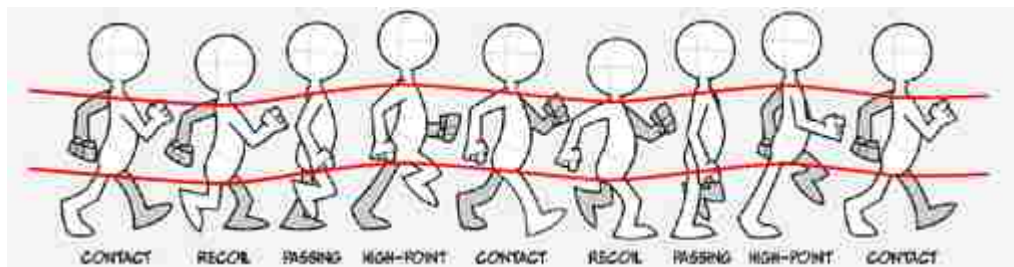


Figure 6: Walking movement in an animation [8]

The camera techniques adopted by beginning animators produce many videos which include jarring movements and poorly selected angles to illustrate the space [9]. This results to distracting animations as well as distorted perspectives of the buildings and landscapes. Aside from a few cameras which align the height and focal length of the lens to a realistic human perspective, many animations still indicate that the classic bird's eye overhead fly thru view is greatly overused and 360 degree spins lack camera and speed limitations. If some of the principles were applied from the research above, such as smoothing out jarring movements by

easing in and out, integrating an overall slower speed and incorporating more pauses in between crucial actions and frames, more consistent animated pieces can be produced. Staging is another area which can also be improved upon in these animations. Although modeling extra architectural components is time consuming, they are a worthwhile effort since they are essential to enhancing an architectural space. Many of the videos had a shortage of these details, resulting to sparsely laid out areas which lacked visual interest.

2.1.3 *Rendering*

Rendering is the process of generating an image from the 3d model. However, due to the use of advanced tools which calculate lights in a scene, as well as material textures and high poly models, renderings become a time consuming process which require intense computing power. The long render times is one factor that influences the decision process for animation length since projects must meet a balance between time and quality. Although render farms, a set of computers which are dedicated solely to rendering, are available commercially, they are expensive. To counter this issue, research was completed on options to optimize renders while balancing quality versus time commitment.

2.1.3.1 Render Selections

Prior to selection of a render to use for a project, a firm understanding of the variety of different methods each rendering software uses for calculation is helpful. There are so many options available to the consumers that it is quite confusing and conflicting to choose a renderer. Usually the solution is to utilize the renderer that comes with the 3d modeling package. Yet it is beneficial to be knowledgeable of the individual and unique features as well the range of controls that may or may not be present in a specific renderer. That way the material selection and lighting process can be specifically selected to maximize renderability and quality in the final product. Therefore renderer selection becomes a choice based on need, features, availability, and the type of results or quality expected for the project. The lack of standard and universal features in rendering systems is a similar problem to the large variety of 3d architecture modeling software available today. In both cases, it is a struggle and effort for the designer both to learn and understand the features available. Unfortunately, until a singular system becomes universally accepted and

meets the needs for the majority of projects, this will be a constant issue within the industries that use these software. For now, a designer must balance the desired features of the tools they select based on the attributes that are of importance.

For example, a deciding factor in renderer selection is based on the qualifications of whether they are physically based rendering systems. There is confusion between renderers which are physically based versus photo-realistic since the ways of determining this is difficult for an end user due to the complicated technical specifications. Photo-realistic renderers place emphasis on the end result of a rendering, its appearances rather than the techniques used to derive it. The use of realistic values for lighting is not a major concern, since these shortcuts allow users to obtain results easily. Physically-based renderers, follow the physical behavior of light as closely as possible in an effort to predict what the final appearance of a design will be [10]. Since architectural spaces become constructed buildings with inhabitants, enabling correct representation of physically correct lighting in a space is important. The level of importance within each project though is still up for debate and dependent based on the overall goal of the specific project and the message desired to be conveyed.

Research and data gathered by [11] in 2010, provided a large list of rendering systems, which included RenderMan, Maxwell, V-Ray, Indigo, and HyperShot to name a few. Almost all of these were photo-realistic renderers, with the exception of Maxwell and Indigo. V-Ray settings determined whether it was physically based or not. A 3rd party validated physically based renderer, which was not mentioned in this list is Radiance [10].

Below is a chart of rendering systems that were candidates for use in the Open Tours project. For this project the renderers were initially selected based on their ability to render in real time, using GPU to obtain feedback on rendering information in order to maximize time and quality. In addition, they were chosen based on their popularity and amount of usage for visualization within the architectural community. The requirement for whether the renderer was physically based or not was not a requirement for the Open Tours project. This was decided based on a number of factors. One, the goal of the project was not based on the lighting evaluation of the final product. Instead, the goal was to use the renderings to explore camera, animation, and rendering guidelines. Two, visualization for the end users are quite subjective especially given

the variation in computer screen resolution and compression quality of animations which would play smoothly on web based applications. V-Ray was chosen based on its prominent presence within the architecture community. In addition, it has the ability to render out different light sets or groups per layer, which was a useful feature to have when manipulating light contribution within a room setting. V-Ray also has a method for rendering based on saved light cache and irradiance maps which became important for productivity in the prototype.

	Maxwell V3	Vray	Indigo Render
Render Times			
not tested, approx due to research	slowest but render quality seems the most realistic	slow	slow
Real Time Features			
realtime viewport	✓	✓	✓
GPU - CUDA or OPENGL	✗	✓	✓
Lighting Features			
selective light contribution extractions	✓	✓	✓
sun and sky for realistic rendering	✓	✓	✓
IES light support	✗	✓	✓
Camera Features			
realistic camera	✓	✓	✓
special camera / effects	SimuLens, stimulate glare effects and can incorporate elements such as dust, water drops, fingerprints		Orthographic / Spherical Cameras for isometric and panoramic effects
Integration			
3d applications	Solidworks, Softimage, Modo, MAYA, Houdini, Rhino, SketchUp, LightWave, Archicad, Bontai, Revit, 3ds Max, Cinema 4d, Form Z	MAYA, Houdini, Rhino, SketchUp, Softimage, Blender, 3ds Max, Modo, Nuke	SketchUp, 3ds Max, Cinema 4D, Maya, Blender, Revit, XSI, XML integration
post production applications	Photoshop, Nuke, AE		
Particle and Effects Integration			
instances	✓	✓	✓
realflow integration	✓		
Special Features			
	3500 ready to use materials	Probabilistic Lights - Increase render speed for scenes with high number of lights	Section Planes cut out rendering
		Render Mask - ability to re-render only selected objects	Uses spectral light computations
		VRmats - material integration between Max and Maya	

Figure 7: Renderer comparisons

2.1.3.2 Pixar Stats

As a side note to encourage and validate the render efforts most designer go through, it is informative to have a comparison based on render times larger companies also face on a regular basis. Pixar's recent film *Monster's University* required a huge amount of computing power to render. Each frame require an average of 29 hours even with the most advance machines today. If only one machine was used, it would have required 10,000 years to finish the entire movie [12]. Given these statistics for such a large animation company, the struggles designers face today in rendering is understandable and real.

2.2 TECHNOLOGIES - TOOLS OF DELIVERY TO THE AUDIENCE

The history of tools and technology used in the past and present are important aspects to research, since knowledge can be gained while learning which portions were successful and the issues that hindered unsuccessful ones.

2.2.1 *3D Authoring Tools*

X3D (2001), is the successor of VRML (Virtual Reality Markup Language) developed in 1995 and a standard file format that is used in representation of 3d interactive vector graphic for the web. X3D is a royalty-free open standards file format that uses XML to represent and communicate 3d scenes and objects [13]. It allows GPU (Graphics Processing Unit) acceleration, meaning it utilizes the user's graphics card rather than the CPU. This is beneficial since it allows for faster and simultaneous calculations based on the threads available in the GPU, rather than singular thread calculations on the CPU. X3D allows for storage, retrieval and playback of real time graphics in embedded applications [13]. A large range of industries have adopted its uses ranging from architecture to scientific visualizations, to games, since it has the capability to support graphics, shaders, CAD data, and animations.

Work in this field have proposed DOM-based integration with X3D and HTML5 to allow directly node access into DOM content, known as X3DOM. X3DOM is an open source framework which aims to run without plug-ins and be part of a HTML5 standard. These tools strive to allow development of authoring tools using XML. Work in [14], seeks to allow easier mapping and

synchronized events with X3D scenes [14]. Their model, a DOM-based integration model uses the integration of DHTML and AJAX , to dynamically make changes to their scenes, and integrate the ability to accept user input for navigation and events. Another project, [15], uses Web3d and X3D to increase environmental awareness by enhancing web communication, increase comprehension and interest of the users. This research is interesting as they explore how the use of interactive environments can be used to engage the users thru the web. The layout of their interface reduces complexity and limits and filters choices for easy interaction. Their application runs a tour based information system that is used to convey information at given times to the user.

These tools stress on the increase possibilities on authoring content on the web and general flexibility in the data that can be customized based on the designer. These possibilities extend the capabilities of a designer by allowing tailored details or component to be easily integrated in a web based application for a specific set of users. In addition, these tools accepts user feedback in determining what elements they wish to view while dynamically changing these elements upon selection rather than a user to designer back to user data exchange that would have had to happen without a web based application. This way users can obtain feedback on specific components of a space instantly. Although the quality of the 3d graphic using this method is not polished and not yet on the level of which a sophisticated rendering can produce, this remains a viable option to investigate further in the future time when technological advancements have progressed.

2.2.2 *Adobe 3d PDF*

Adobe 3D PDF uses a file format, Universal 3D, to embed 3d rotatable objects in PDF documents, and a simple 3d rendering engine to display them. This technology allows the addition of actual 3d models from CAD software to be integrated into PDFs. Users are able to manipulate these objects within the document in limited ways. There are options for customization such as background colors, lighting, and the ability to show existing views [16]. Despite the indications that this format is universal, one has to have Acrobat X in order to make these PDF's. In addition, these models lack integration for complete walk through animations, since they focus primarily

on 3d model display. The concept is notable since the allowance of 3d objects within PDF which are universally viewable can be beneficial to some.

2.2.3 *QuickTime VR*

QuickTime VR, developed by Apple, Inc. was a plug-in for the QuickTime viewer that used images to create a 3d exploratory environment. The images were stitched together to stimulate a 3d environment. Features of the plug-in included panoramic view, creation of movies that allowed the manipulation of objects from all angles, and QTVR scene, which included multiple panoramas and objects [17]. Another feature was the ability to create selectable objects within the panorama that held specific information that was useful. This plug-in focused on the use of rendered or photographic images to develop these environments. Although this plug-in has been retired since 2009 from the developer's library, the concept of using images to develop a navigable space while incorporating the ability to embed useful information per object made this product notable for research.

2.3 CONTEXT VISUALIZATION

2.3.1 *Immersive environments – Game Engine interfaces*

This category of visualization gives users the ability to explore fully immersive 3d worlds. The objects used are full 3d model that render in real time and fully embeds users inside these spaces. Game engines are one primary example which drive these interfaces. They allow import of 3d models into game engines which are customized for user exploration. A couple of issues with this method is the geometry for game engine often need to be simplified and reduce for better imports. In addition there are limits on texture sizes and quality that are necessary since file sizes need to stay within a maximum size in order for the system to run efficiently. Lights are also another issue within game engines, since they are poorly supported and require the use of lights from the game engine themselves which lack the sophistication of lights generated from 3d modeling packages. They do remain an option for users to roam and explore a space.

One example of a tool for architectural visualization tool is called the Architectural Cinematographer described in [18] and [19]. This architecture tool is an implementation of a real time interactive environment that uses the game engine Unreal to integrate architectural concepts and cinematic design in a VE environment. The game engine provides a player with the ability to tour the environment using one of three options for viewing, architectural mode, first person point of view, and free roaming. The architectural mode presents a narrative from the designer while the other two allow the player much more flexibility and control. This tool, in first person and free roaming modes provides a high range of control for the user. This is not beneficial in some cases since the usage of the tool is dependent upon the user's skills and can result to loss navigation in the program or visualization of views that are not meant for viewing.

2.3.2 *Information navigation*

These tools of visualization are used for the delivery of information from the designer to the viewer. These products place precedence in the delivery of information rather than the emphasis on exploration and discovery like the former.

2.3.2.1 Virtual Tour Companies

Virtual tour products provide a medium between game engines and animated videos. These companies provide services in creating virtual architectural tours for clients. They allow for a more focused delivery of information pertaining to a specific space. Most of these tours developed by companies who specialize in the field and lack the ability for designers to tailor the visualization of the space themselves. The ones researched for this paper, [20] [21] [22] [23], all suffer from the similar characteristics of fast moving cameras speeds, especially within the 360 degree viewport. Only [20] provided a plan map which linked the current location of the animation to the map. This feature provides a clear form of orientation for the user and provides them with the affordance that was not found in any of the other sites. Another company, [21], uses guided tours which contain narrated information on the history and development of the architecture as well as the designer. The information provided in the narration was not only educational, but provided an additional connection between the guided tours in reality to their products. This humanistic element provides an element of familiarity which allowed a sense of

connection that can occur between the viewer and the architectural piece. The speed integration for this company also seems to most closely match an average speed of a walking person, allowing less distraction from the architecture and provides the user with a more enjoyable experience.

2.3.2.1 DVIN – DMG

DVIN [24] is a tool which combines the use of coordinated plans and perspective view to navigate building model information. It seeks to convey this information utilizing a person's spatial rather than documentation navigation skills. Developed with Java and VRML, this tool allows easier navigation of information to a wide range of users, not just architects. By utilizing a dual view interface, the tool provided the ability to orient the user while providing information of the space. This combination of information communication enables better understanding and securely reaffirms the location and relative space orientation at a constant basis for the user.



Figure 8: DVIN interface [24]

2.4 CONCLUSION AND OBSERVATIONS

The research above provides a number of observations and information as a starting point in the development of the prototype. One is the concept of providing authorship and flexibility in a given tool for exhibiting architecture. Based on a designer's preference, it is essential to be able

to easily customize and change components of the interface for what is visually displayed to the client. Second is the importance of orientation and awareness of one's location within a tour so that the user is actively aware spatially of their location at all times. The third is a choice made after reviewing current tools and their capabilities. The findings suggested that the information for the prototype would be better present as 2d rendered images and animated clips rather than 3d models embedded within the program due to size and quality limiting restrictions. In respect to the technical aspects, the goal of the prototype is to target a continuous walkthrough within the program that closely mimic the perspective of a person's view point. This was selected (a) based on the challenge of developing a continuation mode that would allow the user to be constantly aware of their location spatially and (b) challenging the ability to generate better visuals that are consistent to one's perspective on ground level, with developed guidelines. It should be noted that the process of generating an architectural model from 3d to rendered image is complicated and time consuming. Given the large number of elements that designers need to learn, the provision of a set of guidelines would provide them extra assistance and reduce the number of elements to consider. In addition, although rendering remains a complex challenge even with today's technologies regarding the scopes of time, computing, and vast array of features and selections, there are methods available in some renderers which can help make this process a bit faster and more efficient.

Chapter 3. OPEN TOURS PROTOTYPE

3.1 CONCEPT AND KEY FEATURES

The prototype is a 3d interactive web based interface which allows users to tour a space. This goal of the interface is to offer a minimalist view with the majority of display window used to display the visual components. This limitation serves a couple of purposes. One, the designer's concepts and ideas become the primary focus on the page. This allows the designer to showcase their concepts, ideas and information. Second, due to the simplified buttons and selection, users are able to navigate and utilize the tool much faster and more efficiently to obtain the information they require.

The features within the software are:

- Interactive interface to engage users and offer easy navigation
- Integration of multiple aspects for the architectural space within on viewing environment: lighting comparisons, 360 degree views, animation, and miscellaneous images
- Applied camera, animation, and rendering guidelines to create more visually consistent and natural looking spaces
- Walk through animation that would orient the user visually and spatially at any given time. In addition, incorporation of a 2d map would extend this concept so the users could be provided multiple perspective of where they are currently within the space.
- Web based and lightweight for easy and portable access

The Open Tours prototype employs web based scripting languages and a XML document to ensure flexibility and customizable content, which can be authored more closely by the designer per project. This choice would allow changes and customizable content later on a per visualization project basis.

3.2 DEVELOPMENT TOOLS

3.2.1 *Software*

3d modeling software:	Rhinoceros, VisualARQ and Autodesk Maya
Animation, camera, textures:	Autodesk Maya
Rendering software:	Chao's V-Ray for Maya

3.2.2 *Hardware*

Render machine specs

Processor:	Intel® Core™ i-7- 3770K CPU @ 3.50 GHz
Installed Memory (RAM):	16.00 GB
System type:	64-bit Operating System
Graphics Card:	GeForce GTX 760 2GB 256-bit GDDR5

3.2.3 Scripts and Languages

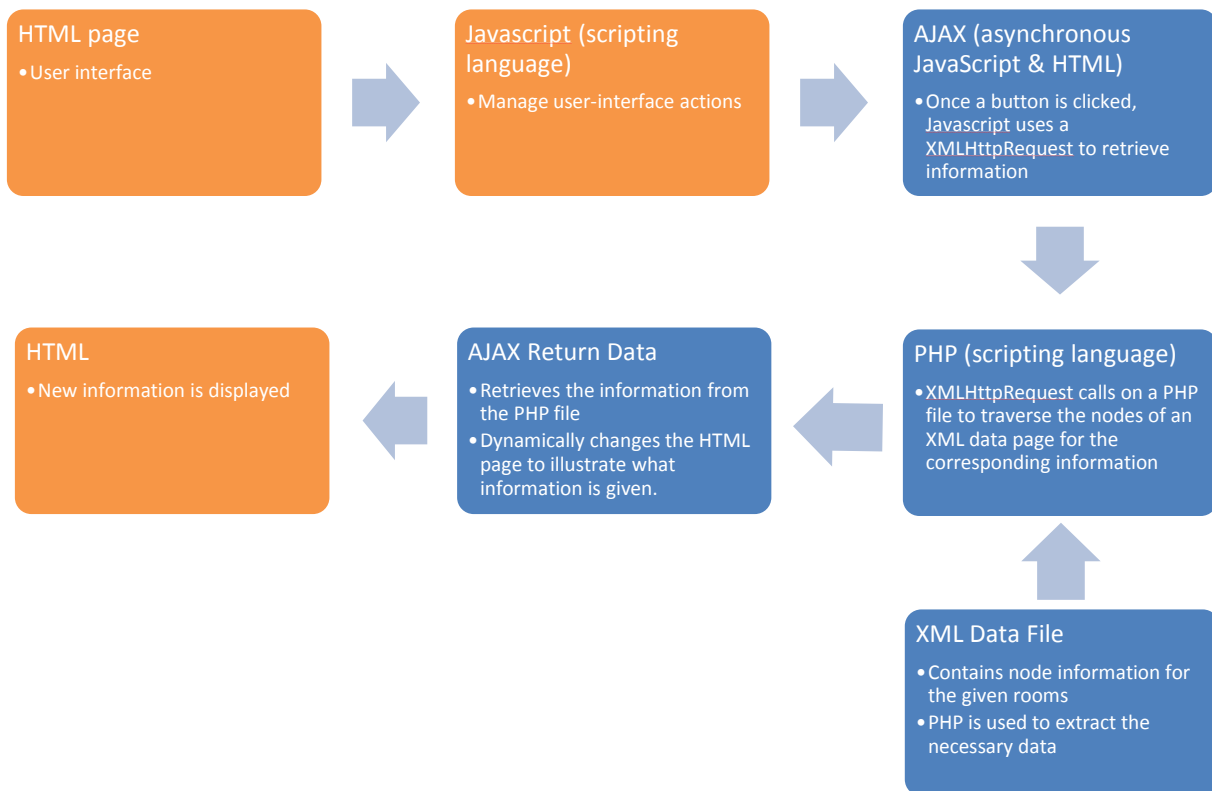


Figure 9: Block diagram of system: front end - orange, back end - blue

The figure above illustrates the languages used and how they connect to each other within the prototype. The pages visible to the end users are shaded in orange. The blue portions of the system works together to update the HTML page for the user based on the options they select. Below is a short definition and use of each language.

3.2.3.1 HTML

HTML (Hypertext Markup Language), the principle language used to created webpages and develop websites, was created in 1991 by Tim Berners-Lee [25]. Written in the form of tags, the code is rendered with web browsers to display visible pages browse-able through the Internet by the user. These pages are the pages that are seen by the user.

3.2.3.2 JavaScript

JavaScript is a dynamic scripting language commonly used to design websites. It allows the website to become interactive based on user actions.

3.2.3.3 AJAX

AJAX is short for Asynchronous JavaScript and XML. AJAX is a set of concepts and technologies for dynamically requesting web content from a server and using this content to update a page [25]. The JavaScript object, XMLHttpRequest is used to retrieve selective data in response to events in the browser, such as a button click.

3.2.3.4 PHP

PHP is a server side scripting language that reduces that amount of redundancy in static HTML websites. PHP pages are dynamically created as needed by the web server. Code for PHP can be embedded within HTML pages, so that only portions of the page are generated dynamically.

3.2.3.5 XML

XML (Extensible Markup Language) is a markup language that encodes data in a text format that can be understood and read by people as well as machines. It allows the storage of customized data between customizable tags, which can be retrieved using AJAX and PHP.

3.3 PROCESS

Below is the detailed process for the development of the prototype. The first section covers the model development process. The second is a recommended list of guidelines for beginning designers to assist them in their animation and rendering process. The third is the web based tool development for the Open Tours interface. The documentation of this process, the issues encountered, and the solutions used to rectify them is meant to serve two functions. One, this is a documentation to illustrate the development of Open Tours and the justification as well as the thought process behind the choices that were used. Second, this document is used to highlight the challenges commonly faced by designers during the 3d development process and increase

the awareness of the issues as well as become a reference point which can be used for learning and knowledge in the future.

3.3.1 Modeling Development and Guidelines

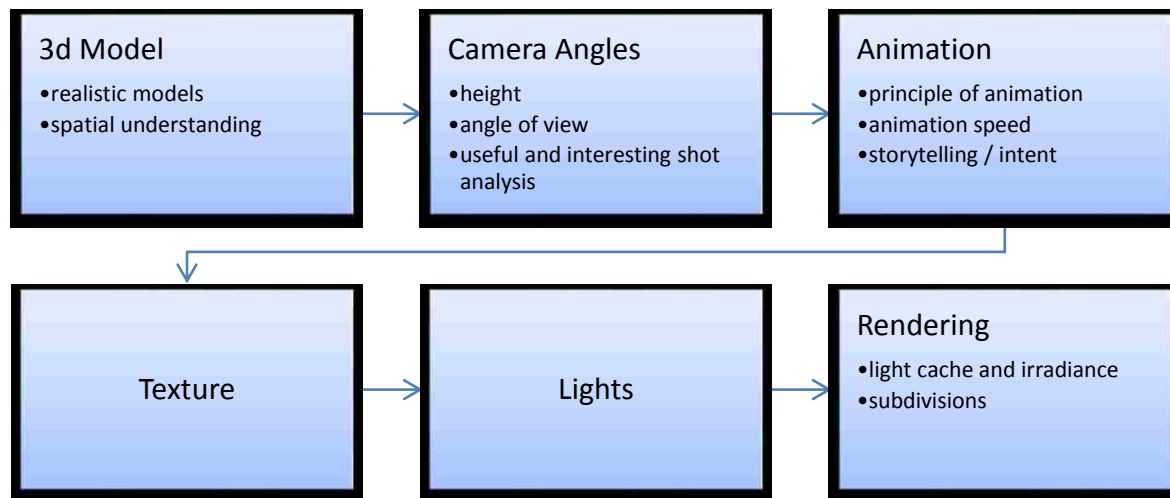


Figure 10: 3d model development steps

The geometry was designed using a couple of software. Most of the structural components and exterior geometry were created using Rhino and the VisualARQ plug in. Although modeling architectural buildings may initially seem easy in comparison to organic models, more experienced designers are aware that elements such as the correct aspect ratio of dimensions in real life to 3d need are vital in order to prevent geometry from appearing out of proportion in the 3d realm. Some designers also use anchors such as people or cars within a space to visually define further the size of a building. The 3d model for this prototype was designed to provide a variety of differently sized spaces for exploration. This way the guidelines can have a larger array of data to examine, such as high ceiling within a small room or large spaces with high ceilings. After the completion of architecture frame of the home, the model was then exported to Maya for detailed work in the interior.

This export process was not a simple one. The struggle of exporting models from one program to another is a common issue designer's face. Each programs treats geometry slightly differently and there are numerous formats for storage as well as exporting. In [26], which was published in 2008, over 140 different file formats were described in addition to a long list of outline errors which occur during the conversion process. This issue is a problematic one since this challenges designers during their design process if use of more than one software is needed. Unfortunately, this is a common case due to the specialized features or unique capabilities available within certain software. Using the roof as an example, the Rhino model consisted of one object while the exported model in Maya became eighteen NURB surface objects. This type of conversion leaves issues in file size, geometry manipulation, and texture mapping.

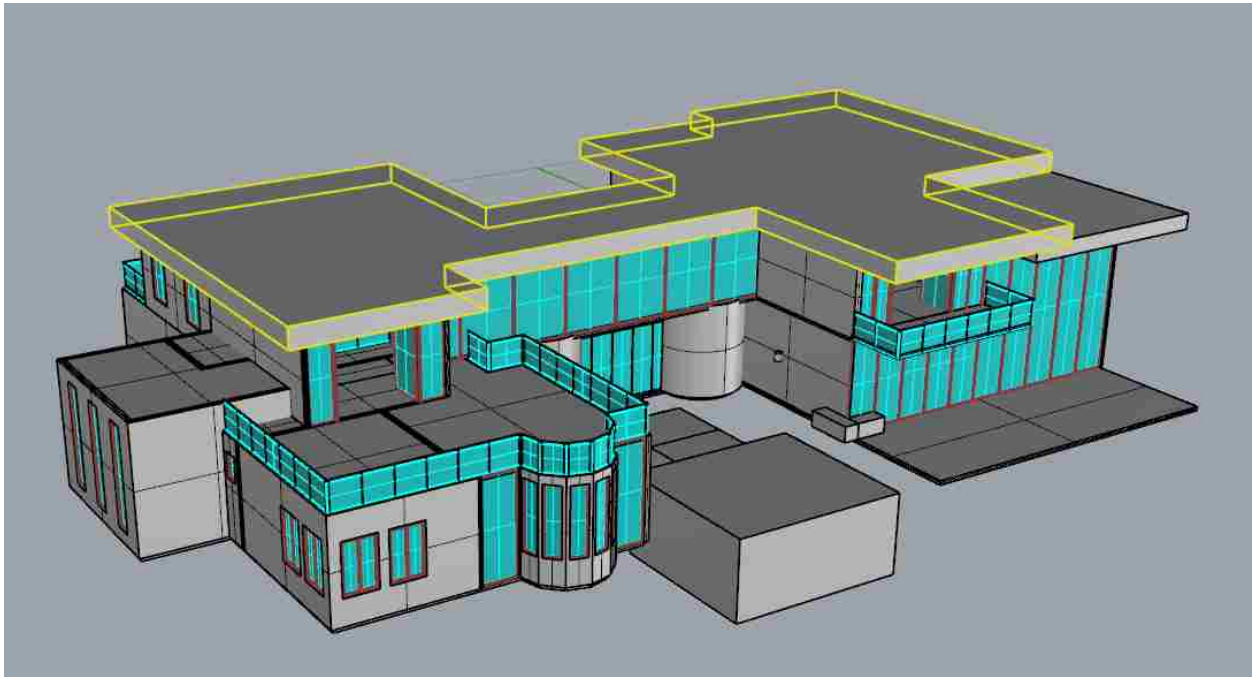


Figure 11 : Rhino model, roof in yellow highlights

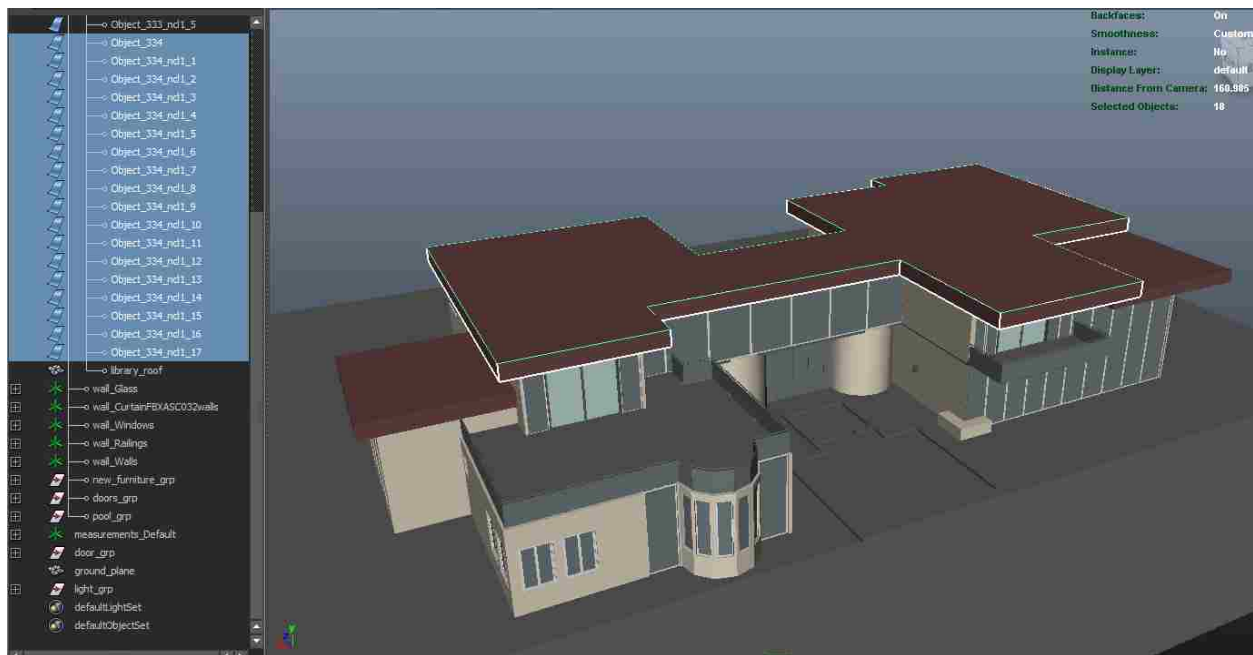


Figure 12: Maya model, roof objects are highlighted on the left

3.3.1.1 Camera

After the experimental manipulation of the camera using the model with concepts from the research, a couple of camera objectives were selected for testing. These were based on their level of importance in respect to the framing of architecture in individual shot compositions and the best methods to illustrate or replicate the idea of a humanistic perspective during a walking tour. Therefore two sections of camera guidelines were developed. One, the evaluation of camera height, angle of view, and degree of rotation based on a walk through mode. Two, useful tips and observations which assisted in the development of visually interesting and appealing shots based on concepts well known within the photography and cinematography community.

3.3.1.1.1 Camera height (walk through mode)

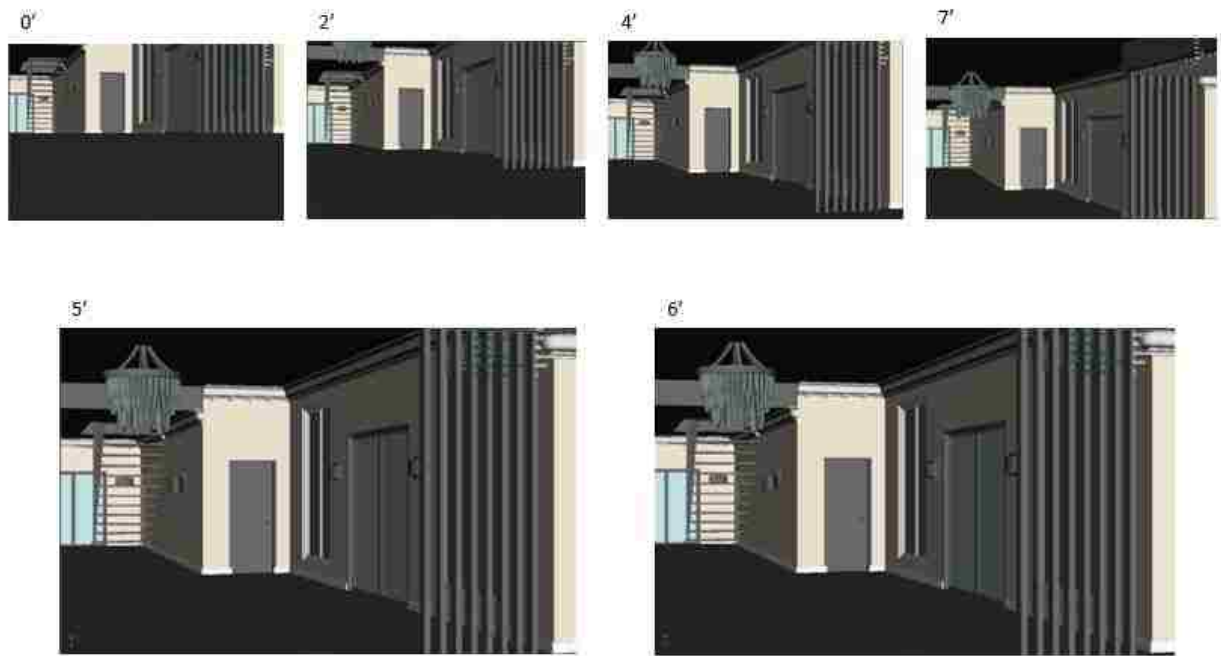


Figure 13: Camera height comparisons

The figure above illustrates a comparison of the camera heights in the scene. Based on the most natural appearance of geometry and viewing height for a walk through, the range of 5 to 6 feet was determined to be best suited. The animations in the prototype used a constant defined height of 5.5 feet in the scenes for the walk through animation.

3.3.1.1.2 Focal Length (walk through mode)

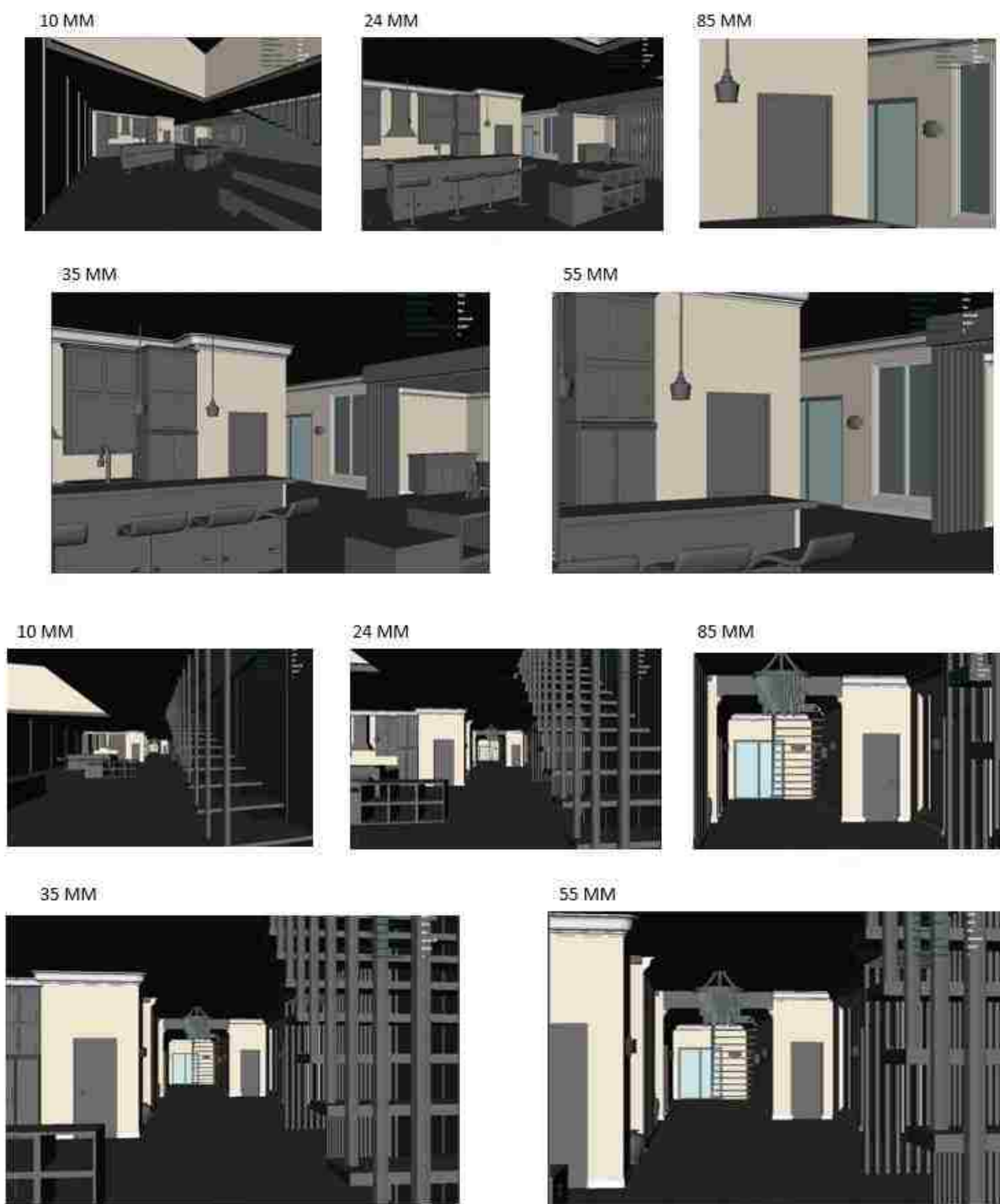


Figure 14: Focal length tests

This focal length test confirmed the findings supported by [2], that the most realistic perspective could be obtained from a focal length range of 35mm to 55mm. Note that this range optimizes the viewing angle, so that objects in the scene appear correctly proportioned based on the camera's or person's distance from the object.

3.3.1.1.3 Focal length 2

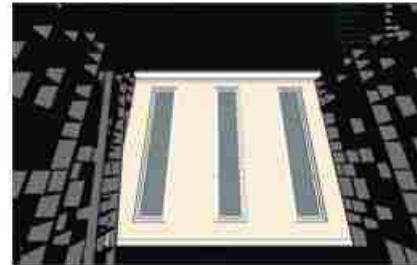
An additional set of tests was performed based on room dimensions, height of ceiling, and focal length. These were used to illustrate and validate the observation that although the recommended angle and heights are good for general cases, other situations, such as high ceiling spaces, or tight dimensions spaced room may require a much different set of settings. The finding reconfirms that most guidelines are applicable within very general settings and should and could be change to define based on a designer's concept or challenges in space sizes.

Library, dimension are approx. 20.5' x 17.5'. Height of walls are 15'
Camera height is at 5.5'

10 MM wide angle, no rotation



10 MM with a 15 degree rotation upward



24 MM wide angle

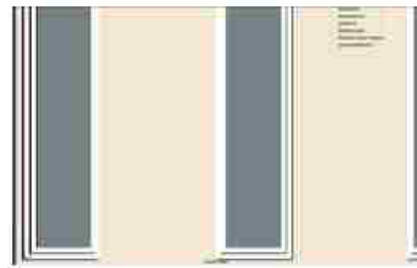


Figure 15: Library settings

The library above requires a much wider camera angle (shorter focal length) to take in the geometry and wall height. Although this perspective is not a natural viewing angle, it

compensates by being able to incorporate an entire scene based on one image. Therefore, in situations such as this, one must weight which concept take precedence, and attempt to balance the image in respect to perspective and perception of distance and accuracy.

Closet, dimension are approx. 14' x 16'. Height of walls are 10'
Camera height is at 5,5'

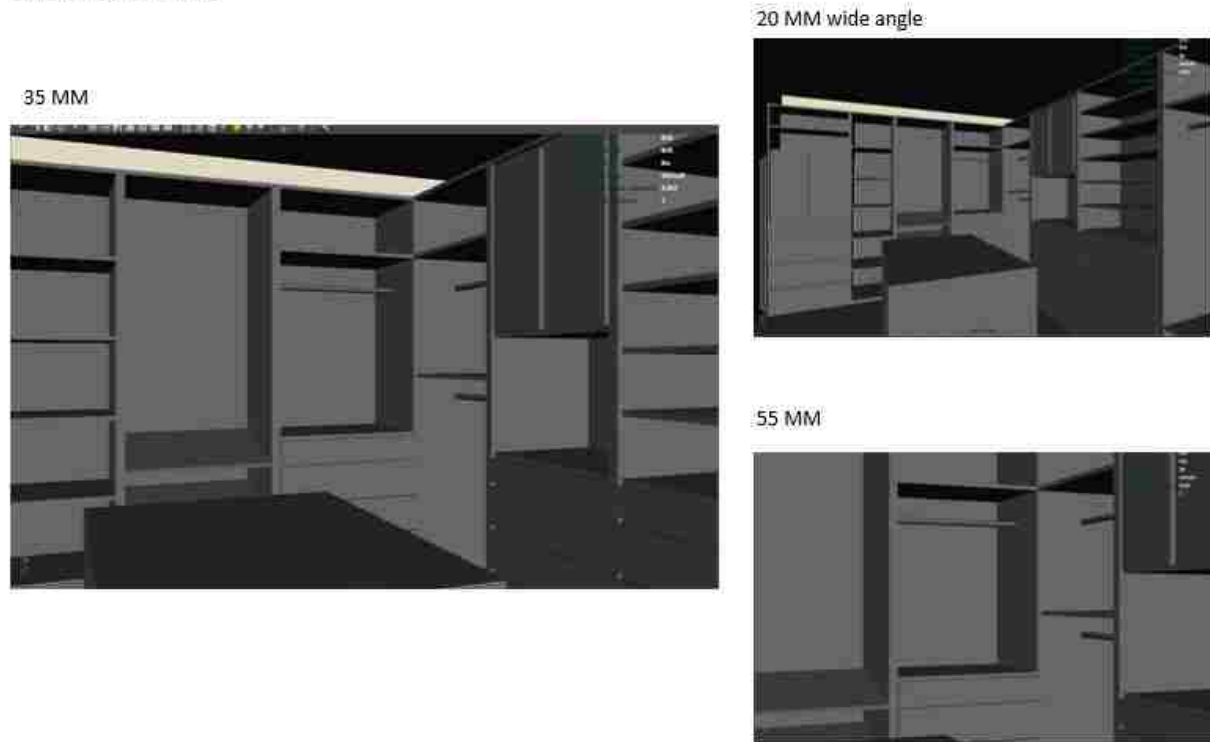


Figure 16: Closet settings

The closet settings above show that 35mm is quite acceptable for optimal viewing. Based on a comparison of these two images, a relationship was made which indicates wider angles become necessary to visually show higher spaces, but result in distortion of height and perception in the size of a room.

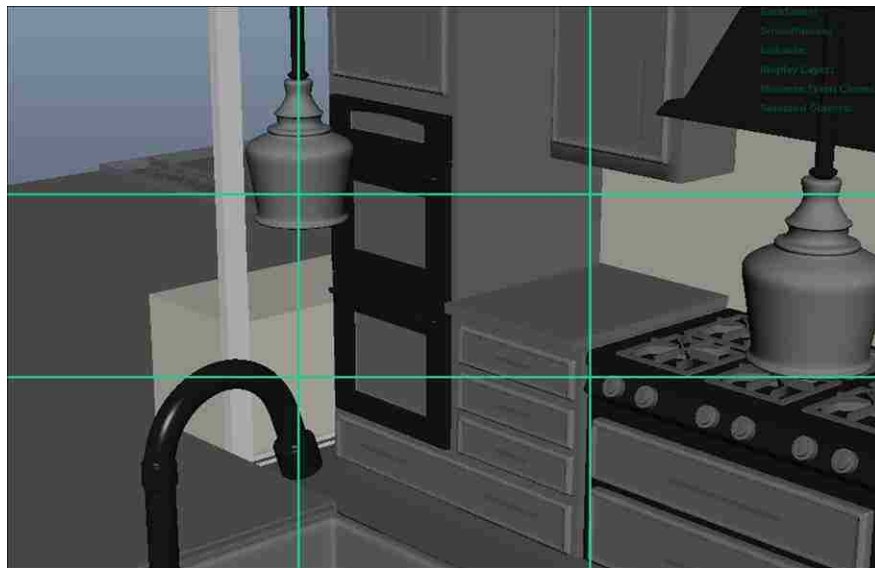
3.3.1.1.4 Angle of rotation (walk through mode)

The angle of rotation for most scenes in the prototype ranged from 0 to 3 degrees. Although a person's eyes are constantly moving, this subtle range of motion is hard to replicate with the camera without causing shaking and unadvised movements within the images. Therefore this slight rotation was set as a norm during the production of the animation, which sought to mimic

and replicate the effect of a person's eyes focused slightly downward while walking straight ahead, given the objective of viewing what was in front of them.

3.3.1.1.5 *Compositional shots*

A range of compositional shots were tested using the concepts in the literature review, rule of third and visual balance. Below is one set of examples which illustrates the different image composition based on alignment following the rules of thirds. The orders of the images are place from top to bottom, most to least in respect to following the rule of thirds layout. The first two is more visually balanced since the distribution of weight is better balance in these images. The last image also appears slightly off centered, even though it is the same image within all three frames, just the cropping is different.



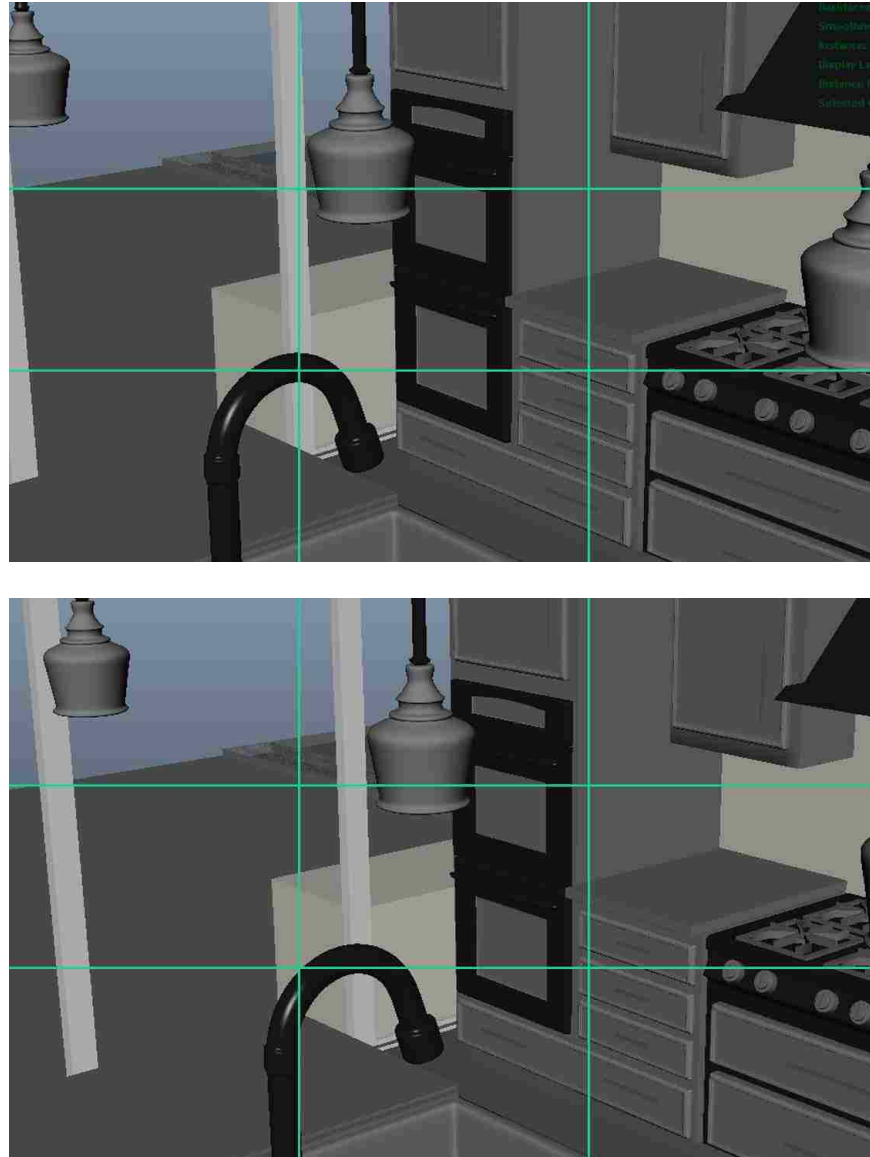


Figure 17: Rules of third comparison

3.3.1.2 Animation

In order to simulate a walk through, walking pace, both in linear movements (in frames per foot) and during rotation (frames per degree) were needed.

3.3.1.2.1 Walking Pace

Anticipation was a crucial element of the literature review for animations, yet it is also a difficult one for a camera to achieve due to its lack of ability in incorporating knowledge of a turn before the action occurs. A person, unlike cameras, has knowledge prior to their movements. In fact, the eyes move ahead of the head turning, since people tend to look before they turn [27]. This is

unfortunately not possible with animation of a camera. A range of animated clips were used to test how other elements could potentially help convey this knowledge to the user. One solution would be to indicate to the user that a turn was occurring, but this was not beneficial due to the large amount of turns necessary within an animation, not to mention the distraction to have text display each time turning occurred. The solution used in Open Tours was to incorporate of a few still frames (hold animation) and slowing down the frames per rotational degree as a satisfying solution.

The results of the test indicated that a minimum linear walking pace of 4 frames per foot and a minimum rotational pace of 1 frame per degree was necessary to achieve a consistent movement suitable for viewing. This definition was used based on the consideration to maximize reduction of frames based on render time. In the prototype, most motion was actually extended by 1 frame in the walking pace and 1.5 frame per degree within the rotational pace if render times could be accommodated.

3.3.1.2.2 Ease in and ease out

Animations should incorporate easing in and out frames of motion to smoothly replicate natural movements. Note the two graphs below for comparison used to illustrate a rotation on the Y axis. The first one has smooth wave forms that flow from one high to another. In comparison, the second shows jagged edges in the peaks. To obtain smooth flowing motion, animation graphs should visually mimic sinusoidal curves rather than mountain formations.

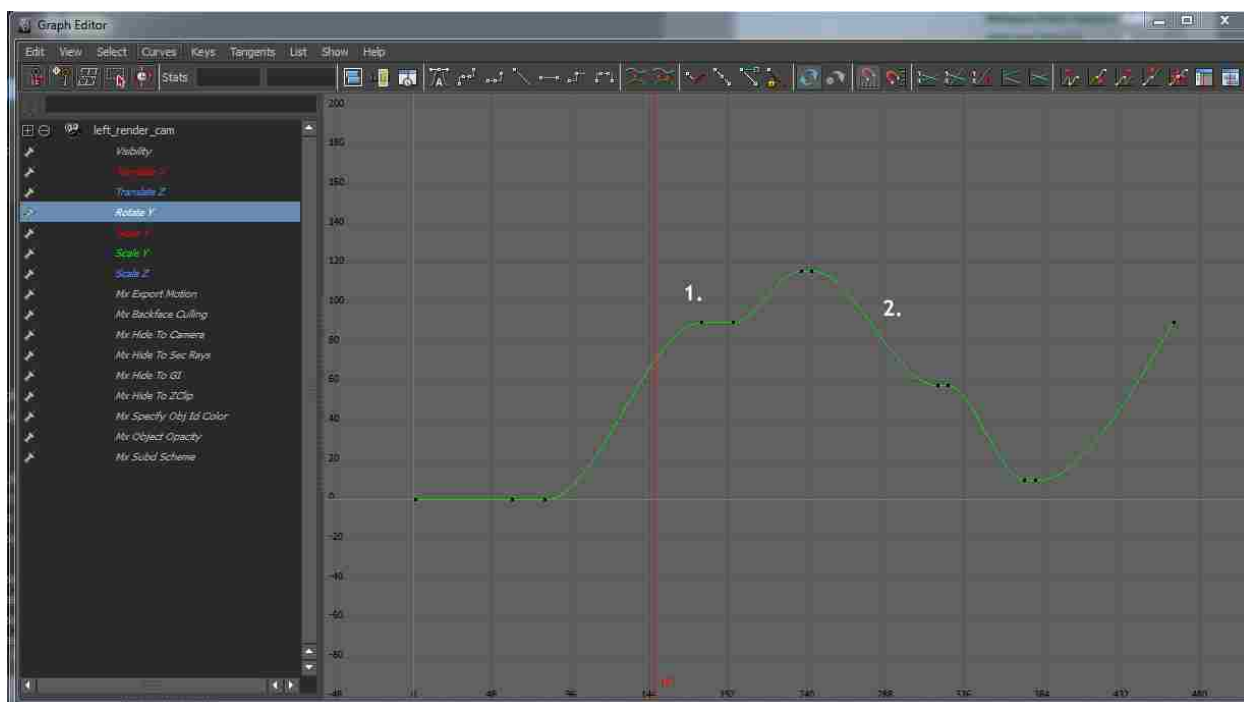


Figure 18: Smooth flowing animation graph appearance

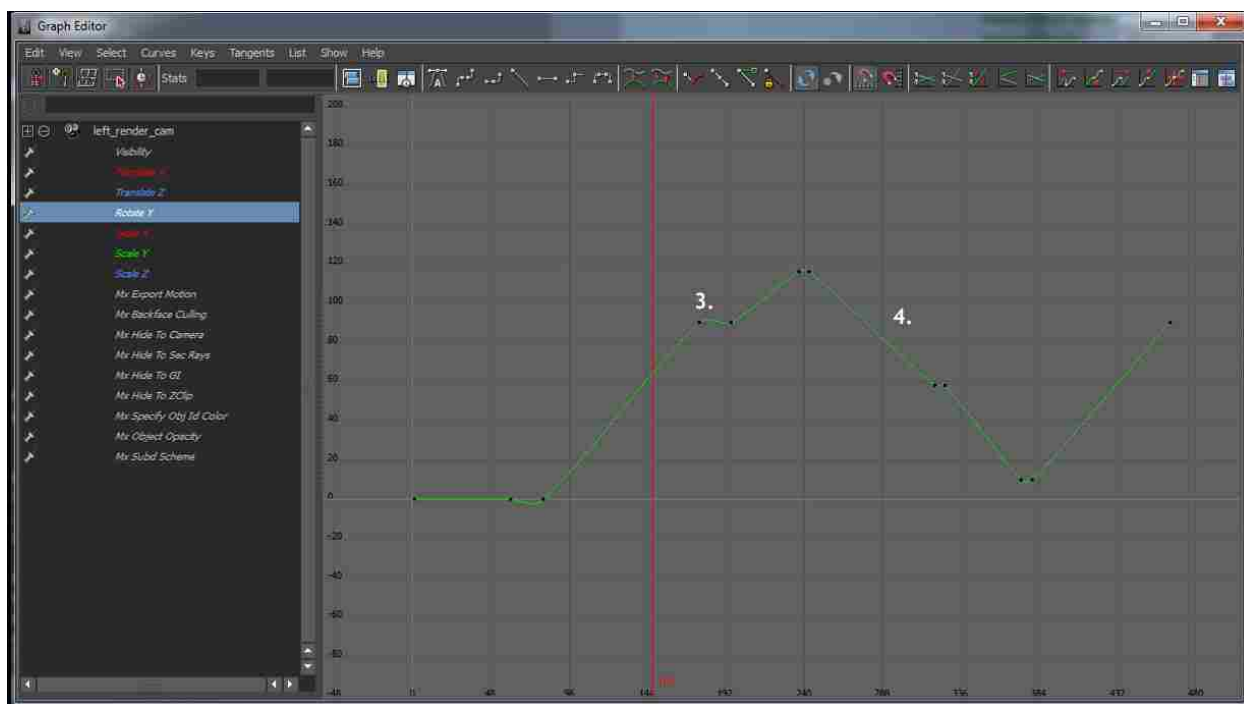


Figure 19: Incorrect graph appearance

3.3.1.2.3 *Timing*

Timing is an integral part with the combination of anticipation, easing in and out and staging in order to direct the viewer's attention towards a designer's intentions [27]. Proper timing is crucial in allowing ideas or concepts to be conveyed to the audience. Designers of architecture must be conscious of the amount of time spent in a room, so that the necessary details and parts are shown without speeding through or lingering too long. In addition, the camera should pause and / or stop for a few frames to break up rotational or walking movements. Ideally the stopping points should also target framed visual content so that the most impactful moments absorbed by the viewer are interesting and well balance compositional views of the space.

3.3.1.3 *Rendering*

Rendering is a constant challenge even in industry where computer power is much more readily available than compared to small firms and individual designers. In addition to rendering settings being complicated and confusing, frame calculation are dependent upon a large number of factor, such as texture, lights, geometry, and special effects. For the designer of architecture, obtaining realistic textures and lighting environments often results to an increase in render times due to the heavy calculations required for interior homes. Although rendering is not an issue that can be solved without either increasing computing power or time, factors that are usually not available to the designer given cost or time allocated for the projects, there are options within some renders to reduce and save time while still obtaining suitable quality work.

3.3.1.3.1 *Light Cache and Irradiance Maps*

Within the V-Ray renderer, one solution which reduces render time is a three part method which applies saved light cache and irradiance maps in renderings. By reducing the need to calculate this information at every frame, the result is in reduction of render times.

Light cache is a method developed by Chaos Group for the V-Ray renderer for approximating the global illumination in a scene [28]. Light cache is built by tracing a large number of paths from the camera and stores the illumination in a 3d structure, potentially independent of the camera position. Light cache calculation has the ability to reuse a sample taken by another prior trace rather than resampling, which helps to speed up the calculation process. Below is a diagram of how light cache works.

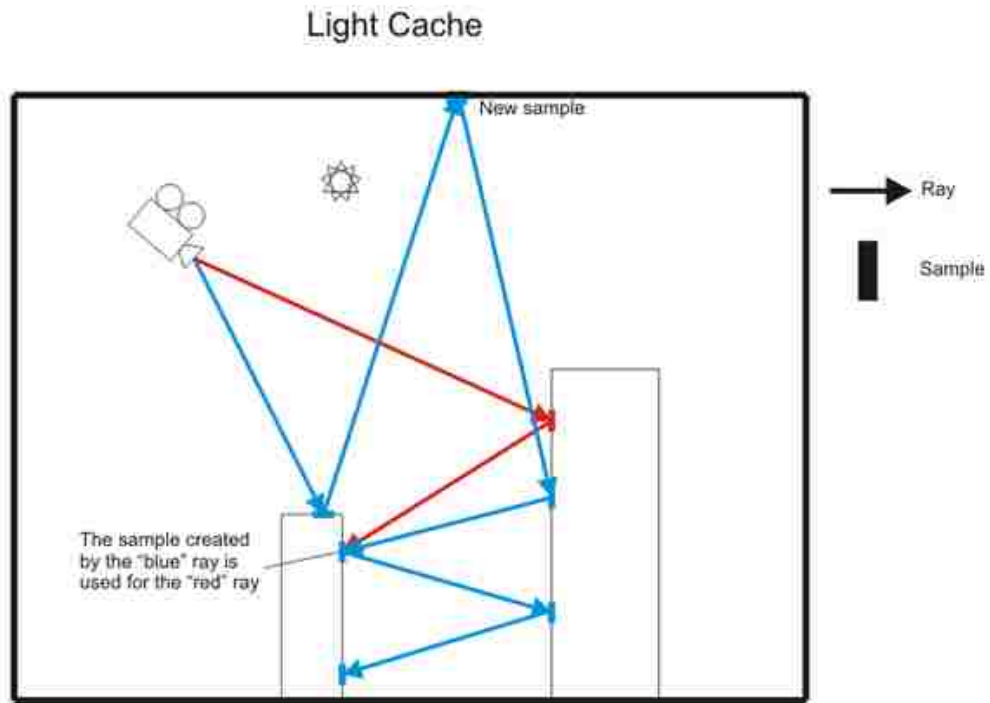


Figure 20: Light cache illustration [28]

Irradiance is a function defined for any point in 3d space and represents the light which arrives to this point from all directions. Irradiance maps are a collection of points in 3d space (a point cloud) along with the computed indirect illumination at those points [28]. Below is the irradiance map calculation diagram.

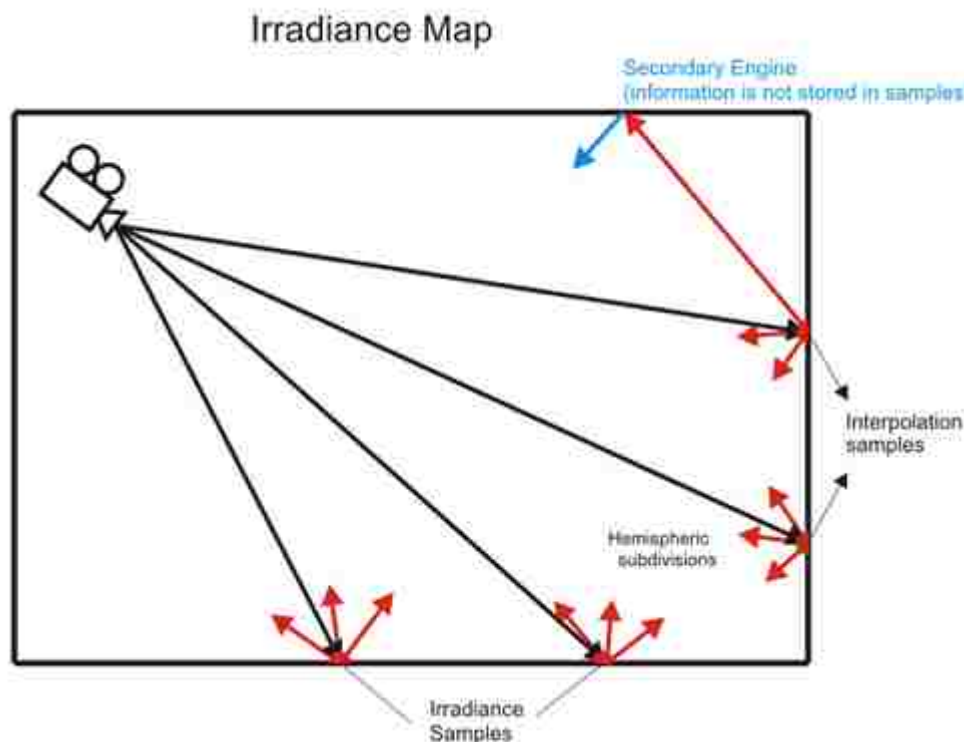


Figure 21: Irradiance Map [28]

Both methods are view independent which makes it possible to use saved maps for rendering. This method is perfect for static architectural walk through when the camera is the only thing moving in the scenes.

The first step of the three part method is to render the animation using the fly through mode setting under the light cache section of the Global Illumination (GI) tab within Maya. This fly through mode calculates all the frames in the visible timeline. The map for the light cache is then saved. The second step renders the irradiance map as the primary bounce engine and using the multi-frame incremental mode setting for the entire animation. Once these two maps are saved, they are used to render the actual animation. Using these setting in one test case animation consisting of 155 frames the save render time amounted to over 6 hours. Given that this percentage was approximately a reduction of 40% in render time, this method should definitely be considered, especially if the scene is static except for cameras and there is a time limit placed on the completion of a project. Most of the Open Tours animation renderings were achieved using this method.

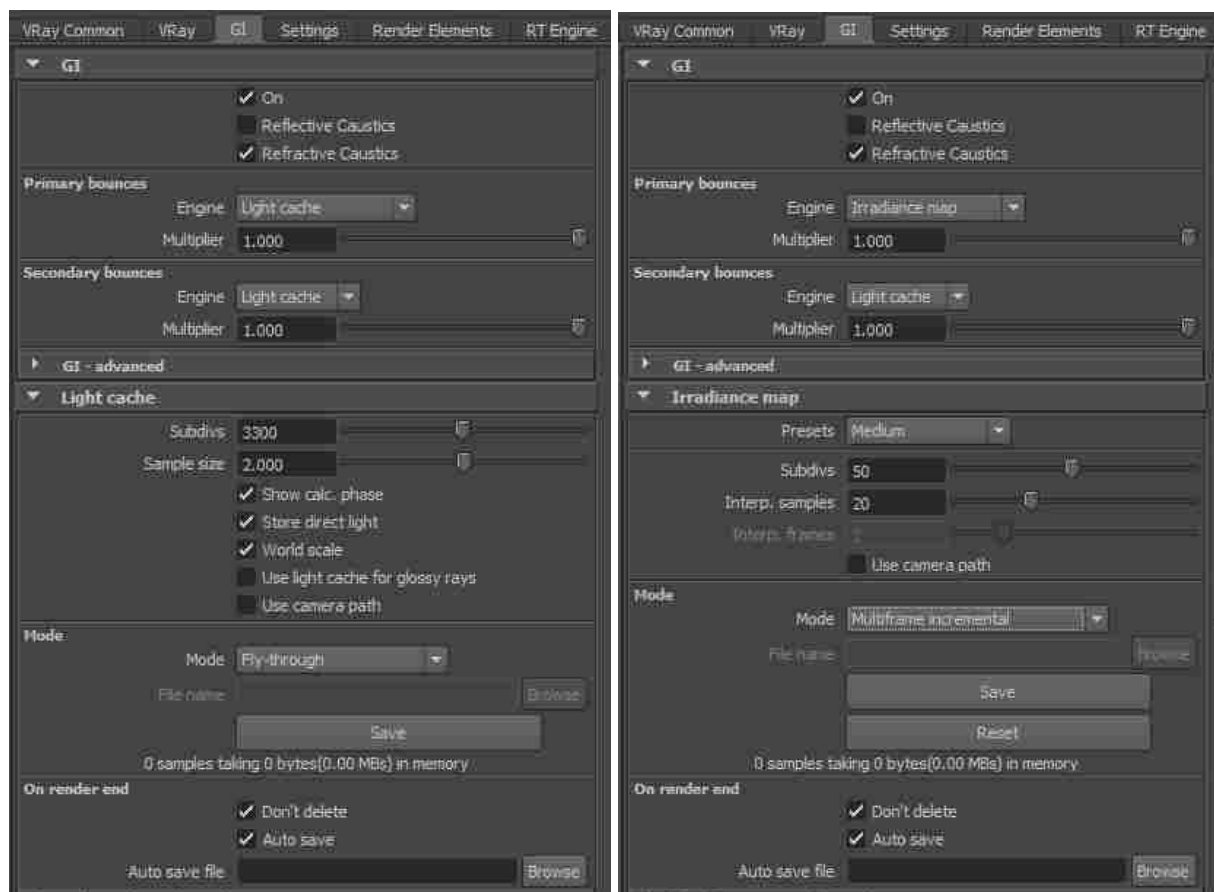


Figure 22: GI tab in Maya, Light Cache (left) and Irradiance (right) settings

This method had a few drawbacks. One is that the calculations place more weight in the information gained by the first frame of the animation. This results to different variation of light shade dependent upon the initial frame calculations. Therefore it is beneficial to render a continuous animation together rather than splitting the frames apart. This become problematic during doors or window animation within these frames. In addition, memory and CPU requirements necessary to calculate the irradiance maps can be quite intense in larger scenes or animations, resulting in memory overload or CPU's which are not able to handle all of the render information.

Despite these issue though, the saved render time outweighed the issues faced due to lighting in this project. Even with re rendering and corrections from light variations, the renderings were still completed faster than use of the brute force method. One other useful

element was that the saved maps could still be used with after material changes occurred. This provided even more flexibility in changing the appearances of a series of images relatively easily, without too much visible difference.

3.3.2 *Recommended Guidelines and Tips*

Below is a list of recommended guidelines developed during the development of Open Tours. The information noted below is subjective and are composed of two sections, a walk through set up recommendation and overall tips and techniques suggested in order to produce visually appealing images. As seen during the testing of camera focal length, these guidelines are general and should and can be altered in order to support specific concepts of a designer.

Walk Through Guidelines		Used in prototype tool
Camera	Camera Height: 5 to 6 feet	Height was locked at 5.5 feet for walkthrough
	Focal Length : 35mm to 50mm.	In the prototype, the animation was composed using 35 to 40 mm range. 50mm was not used as it was not necessary to zoom in that close to objects.
Animation	Linear walking speed: 4 frames per foot, minimum	This speed was extended 1 frame per foot when render time allow for slower movement
	Rotational speed: 1 frame per degree of rotation, minimum	This calculation was extended to 1.5 frames per degree of rotation when render time allowed.
	Hold Frames (anticipation and staging): Include 3 to 5 frames of static animation at visual hotspots during long animated clips, especially important during large rotational views	Static frames of 2 to 5 frames were incorporated into the prototype, used more frequently during 90 degree plus rotations.
Rendering (V-Ray Specific)	Timing:	The timing used in the prototype strived to ensure a smooth animation so that users could have time to visualize the space presented without lingering too long on any one aspect.
	Light cache and irradiance method: If rendering static scenes where only the camera moves, one should look into options to use this method to save render time.	The 3 part method was quite successful during the development of the prototype, and used for most of the animation.

General Tips and Techniques		Used in prototype tool
Modeling	Dimensions and sizes: A worthwhile task is to model objects in their correct dimension. This allows visually correct scene display and makes obtaining data for the objects that much simpler.	In most cases where the object is relatively visible within the scene, they were modeled to have correct dimensions and sizes.
	Details in the interior scenes: Although complete furnishing are not necessary or desired within all spaces, having architectural furnishing such as molding, columns and beams at a minimum allows the space to look more realistic.	
	Polygon count reduction (V-Ray specific): In real life, most objects do not have perfectly straight edges. See “rendering” under Round Edges for more details on how to reduce poly count.	
Camera	Techniques for better composed shots:	Remember to use the rule of third and visual balance to make shots more interesting, especially at stopping points in an animation. Depth of Field should be used to draw focus to a particular element of interest.
Rendering (V-Ray Specific)	Round edges attribute: Render objects using “round edges” attribute, since most objects in reality do not have perfectly straight edges	Most objects within in the prototype either had smoothing or rounded edges on most material to produce a more finished appearance

3.3.3 Interface design Elements

The last part of this project consisted of the development of a web based navigational tool which would play animations in sequential order provided a user selection. This tool would include integration of other information which would be useful for a user to be able to visualize. Although the options within Open Tours is lighting comparison, still images, and 360 degree views, the variety of information which can be included are endless and easily customizable given the XML integration.

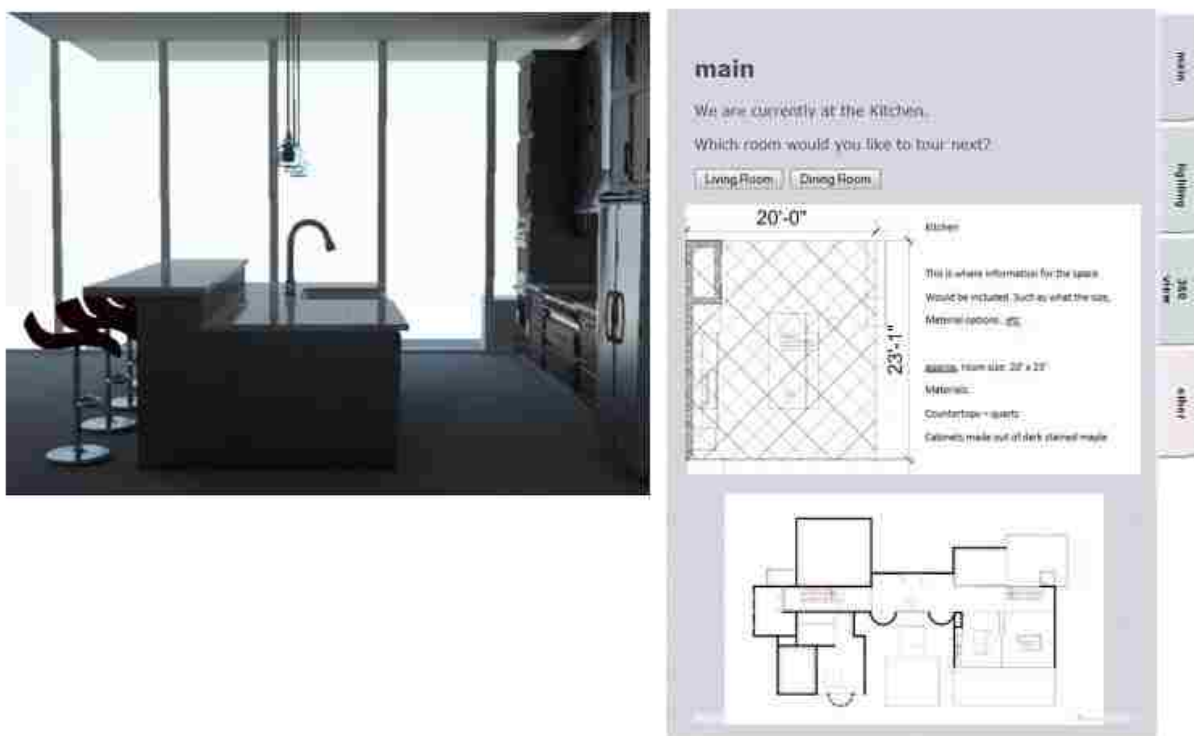


Figure 23: Initial prototype design (version 001)

Above is an image of the initial conceptual design of the tool. Written in HTML, this page is what is viewed by the users on the web. The left side (content holder) and the right side (content manipulator) combine to allow interactive control of the viewing environment. The content holder section is composed of multiple hidden divs of data related directly to the content manipulator tabs (main, lighting, 360 degree view, and other). All of this information is stacked in the content holder section hidden from view until the user selected the related button

associated with that layer. This is possible due to the usage of JavaScript with event listeners on the tabbed button. Users select the information they wish to view and the information is changed dynamically based on AJAX calls to a customized PHP file which extracts the data needed from a singular XML file.

The XML file is the core information holder; it houses all of the information for this project. Below is a sample of a portion of the XML file. In this example, the information is relevant to the kitchen in the prototype. Each of the tags within an XML file is customizable and can be tailored to hold whatever information is needed by the designer. Initially, the interface was designed to only provide a binary set of options at each step. Note the left ("Living Room") and right ("Dining Room") node tags, which indicate which rooms are next on the walkthrough.

```

- <node id="3">
  <location>Kitchen</location>
  - <left nodeid="5">
    <name>Living Room</name>
    <image>images/kitchen.png</image>
    <video light="0" view="1" other="1">video/livingRoom.mp4</video>
    - <lights number="3">
      <light lightgroup="counterLight">images/lighting/kitchen_counter.png</light>
      <light lightgroup="diningLight">images/lighting/kitchen_dining.png</light>
      <light lightgroup="frameLight">images/lighting/kitchen_frame.png</light>
    </lights>
  </left>
  - <right nodeid="6">
    <name>Dining Room</name>
    <image>images/blackWhite.png</image>
    <video light="1" view="1" other="1">video/diningRoom.mp4</video>
  </right>
</node>

```

Figure 24: Kitchen XML (version 001)

Xml layout



Figure 25: XML layout (version 001)

This concept of binary choice became problematic once a completed spatial graph was developed for the space. Note the complicated structure above. In order for the goal of a continuous walk through to be accomplished, the network of nodes became quite large. This, in addition to massively increasing the number of renders needed, also makes for a large amount of repetition and confusing with slight variations of similar nodes. For example, the information necessary for the kitchen node is repeated 4 times in slight variations. This makes an already complicated rendering 4 times as complicated.

The layout for version 001 was also not efficient based on the goal of the project. With a concept of a visualization tool, it was wasteful to use over 40% of the space for a tabbed menu and text content.

3.4 FINAL PRODUCT AND FUNCTIONALITY

In order to change the issues present in the initial prototype, a second version of the tool was developed. The individual components of each section will be explored in detail below.

3.4.1 *Overview of HTML page*

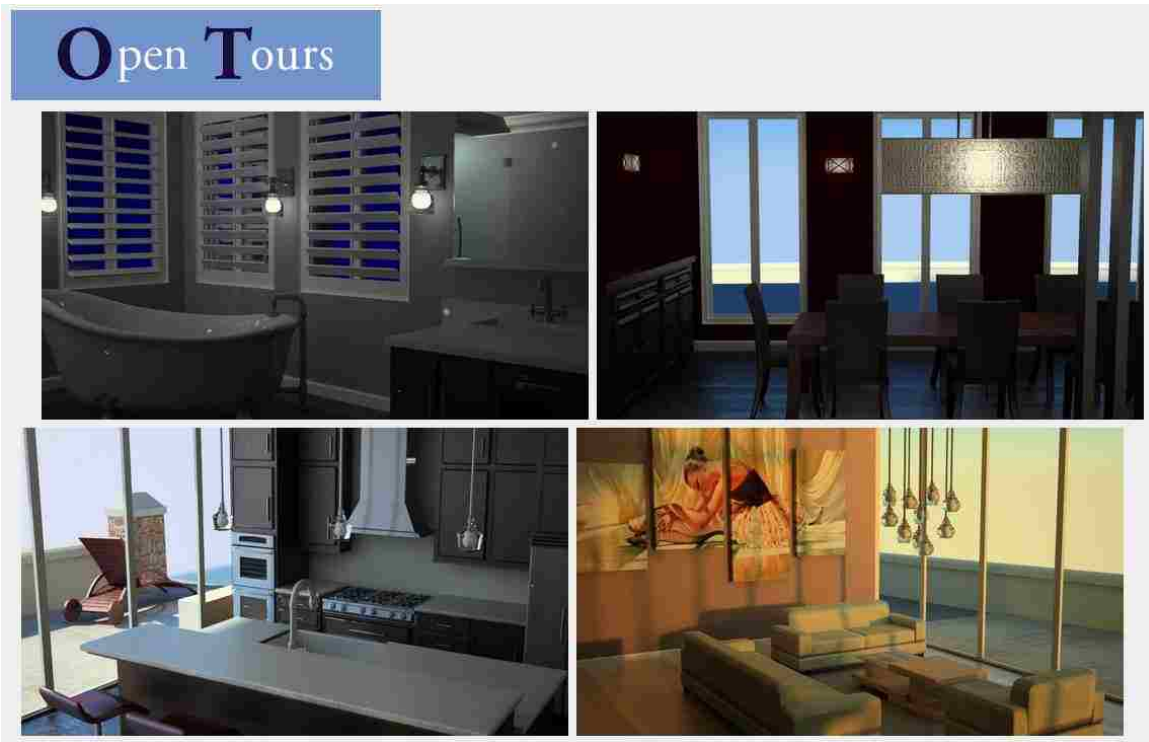


Figure 26: Welcome Screen

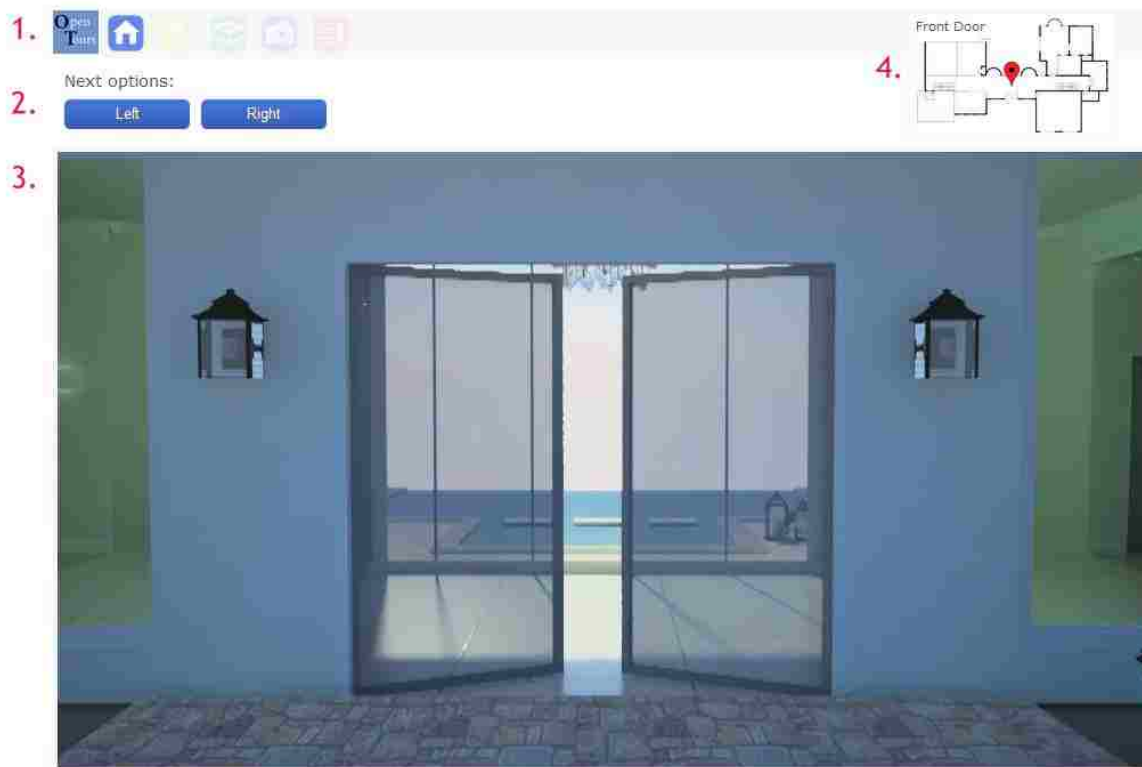


Figure 27: Main Screen viewers see when starting the tour

The image above illustrates the initial welcome screen and opening animation for the prototype. All the information is dynamically driven and controlled by the retrieval of the current XML node with the event listeners and AJAX calls. The numbered sections are described as follows:

- #1 Buttons, noted in sequential order as home, lighting, 360 degrees, images, and other. These buttons are enabled and disabled based on the XML information retrieved. These buttons set and control the visibility of section #3.
- #2 Navigation options, dependent upon current location.
- #3 Visual layers of information stacked on top of each other. Each layer's visibility is connected to the button in section #1.
- #4 Map information based on current location.

3.4.2 Node information

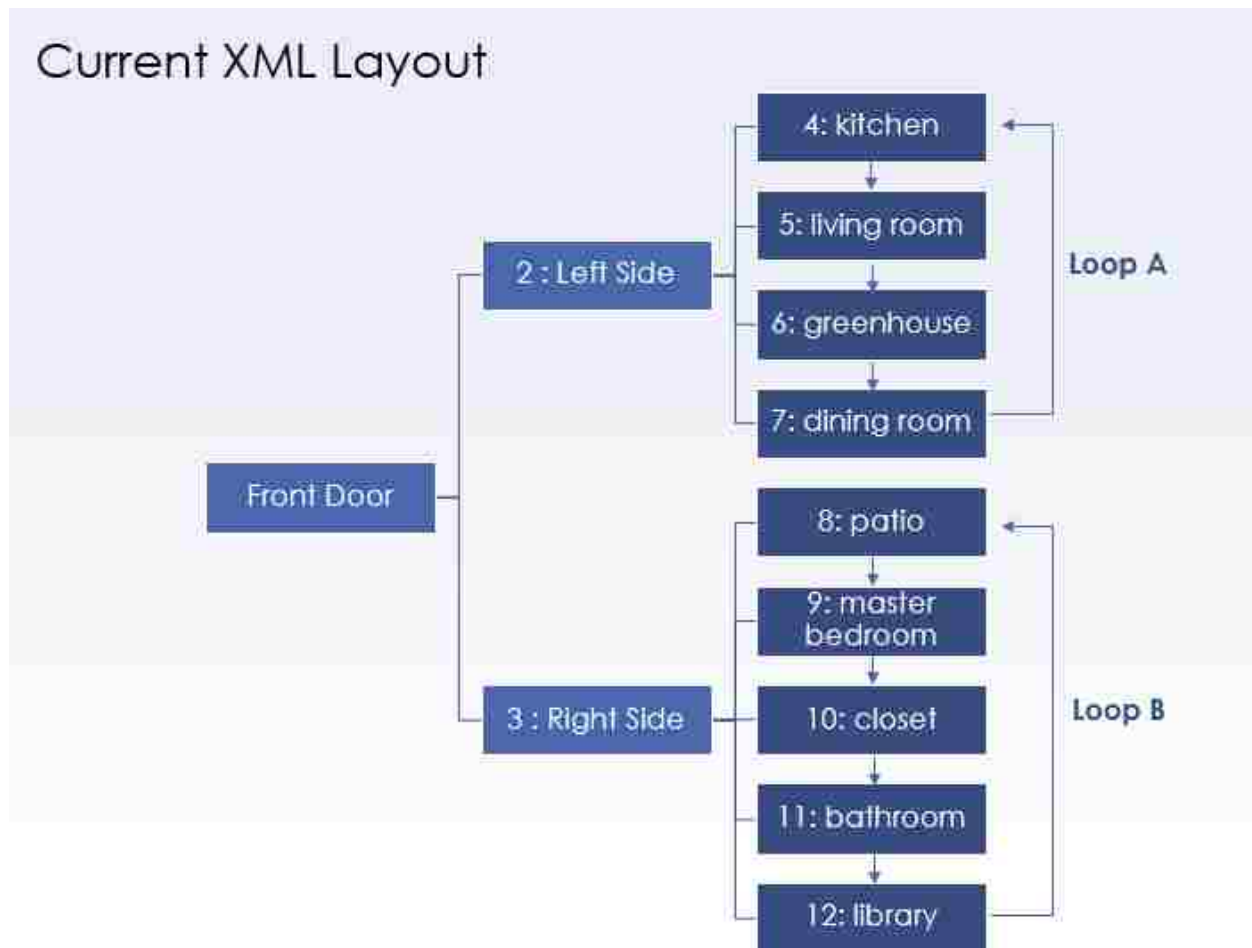


Figure 28: Reconfigured node information for XML

The node information for the XML was reconfigured to reflect the network shown above. By changing the options of each location to incorporate a “back button” and limiting access to one room at a time, the amount of animation and rendering required was reduced, while still allowing users to visualize the space in on continuous walkthrough or in individual components, as desired. The intent and information the designer wants to illustrate is not limited by complexity since the levels can be easily added or removed. User would benefit from having a variation of possibilities for exploration. The integration of a circular loop of rooms (see loop A and B) allows for reduction in the animations that need to be rendered as well as allow one to

leverage the use of light cache and irradiance capabilities while guaranteeing continuity while moving from segment to segment.

Below is the XML data node for the bathroom and the pages which are generated given the provided information. A breakdown of the individual pages follows.

```
- <node id="11">
  <location>Master Bathroom</location>
  <buttoninfo light="0" view="0" other="1" picture="0"/>
  <video0>video/bathroom.mp4</video0>
  <video1>video/bathroom.mp4</video1>
  <image>images/kitchen.png</image>
  - <lights number="5">
    <light lightgroup="bathroommorning2" title="Morning light, all lights on">images/light/bathroom_morning2.jpeg</light>
    <light lightgroup="bathroommorning1" title="Morning light, no electric light">images/light/bathroom_morning1.jpeg</light>
    <light lightgroup="bathroomnoon" title="Afternoon sun, no electric light">images/light/bathroom_noon.jpeg</light>
    <light lightgroup="bathroomevening1" title="Evening setting, All lights on">images/light/bathroom_evening1.jpeg</light>
    <light lightgroup="bathroomevening2" title="Evening setting, wall lights only">images/light/bathroom_evening2.jpeg</light>
  </lights>
  - <pictures number="2">
    <picture picturegroup="bathRand1" title="Facing Mirror">images/pictures/bathRand1.jpeg</picture>
    <picture picturegroup="bathRand2" title="Sink and Countertops">images/pictures/bathRand2.jpeg</picture>
  </pictures>
  <map left="140px" top="-110px" leftLarge="735px" topLarge="-450px"/>
  <threeSixty id="masterThreeSixty"/>
  - <next nodeid="12" videoPlay="video0">
    <name>Library</name>
  </next>
  - <next nodeid="3" videoPlay="video1">
    <name>Go Back</name>
  </next>
</node>
```

Figure 29: XML data for bathroom node

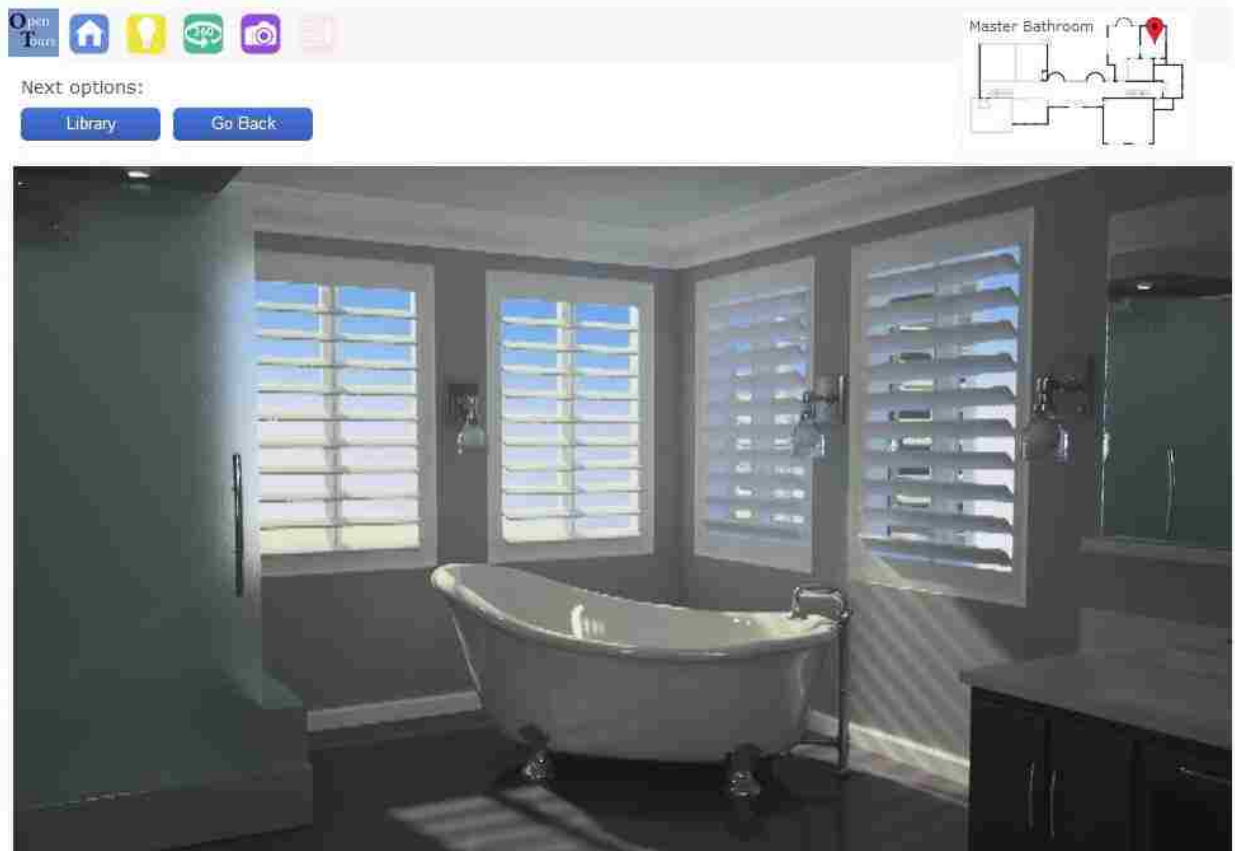


Figure 30: Home screen

3.4.2.1 Home Screen

This home page is the section which plays all of the walk through animations for Open Tours. It is unique since it offers the concept of a continuously oriented animation in a personal walk through perspective. The home page uses the `<video0>` and `<video1>` tags to specific video locations. The purpose of the `<video0>` and `<video1>`, is to differentiate between locations the camera is coming from previously. This is important since it is what minimized the connections that were needed in the version 001 XML chart. The information in the `<next>` tags fill in the locations dynamically based on what room is available to view next. Buttons on the top of the home screen are deactivated until a video is completely finished. In addition, when one navigates to a different layer, the “next option” buttons are disabled, so that one can only proceed to a different location if they are on the home screen.

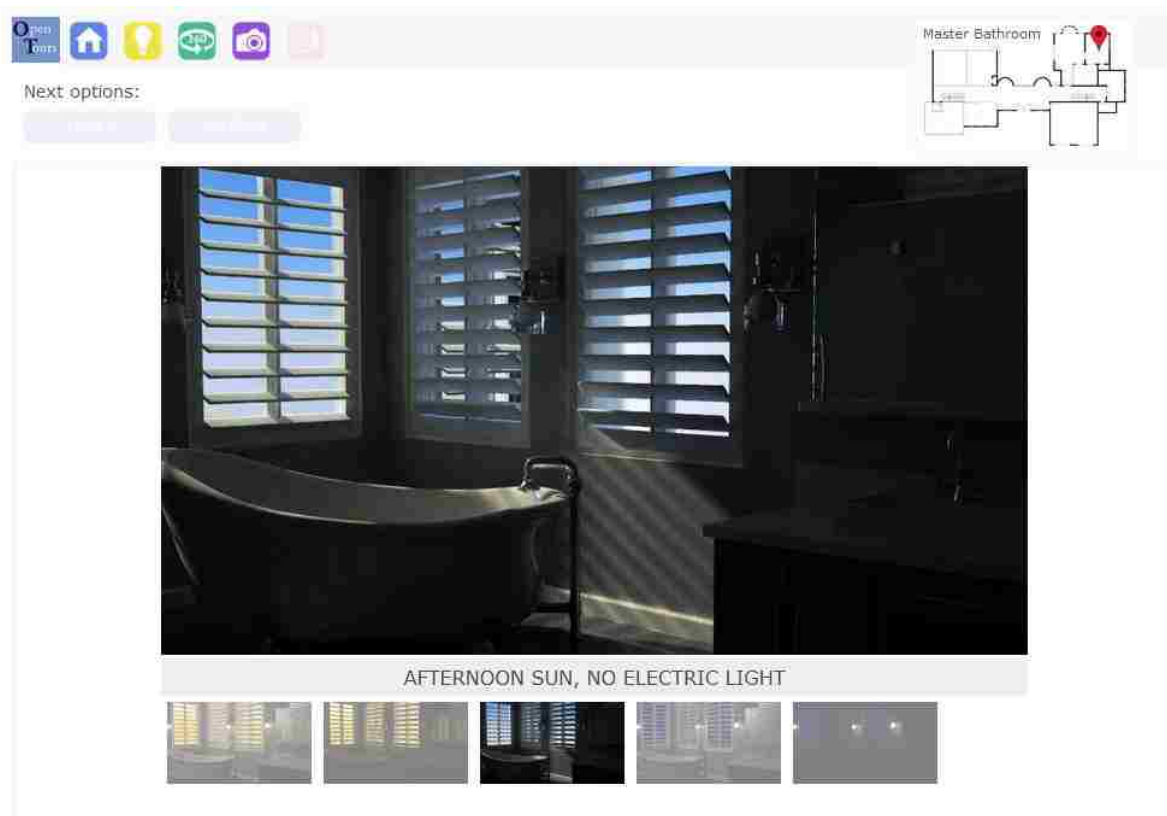


Figure 31: Lighting screen

3.4.2.2 Additional Layers of information

Unlike the fixed perspective of the home / animation screen which used a person's perspective viewpoint, the other buttons within the prototype provide information using a variety of different perspective. This is useful for a number of reasons. One, people understand and gain information based on variety of different methods for delivery. This reasoning is supported by the DVIN project, which used dual views to maximize user orientation. By offering information in different visual perspective, it increases awareness and ability to deliver information to the user. Second, provision of an architectural space using views not seen by a person's ground perspective is extremely useful in order of a user to gain overall and detailed images of a space.

3.4.2.2.1 Lighting Screen

The purpose of this screen is to present different lighting scenarios for a specific space the user is visiting. The screen consist of identical images which vary based on lighting scenarios. The information for these images are held inside the <lights> tag. Each <light> tag holds information

for a specific image. The entire group is displayed as thumbnails as noted in the “Lighting Screen” figure. Note the individual lighting files in the xml titled “lightgroup” and “title”. The “lightgroup” attribute is used to dynamically provide a CSS class and id to the particular image so a slideshow effect can be displayed when the user selects a particular thumbnail to open a larger view of the selected image as seen above. The “title” attribute holds the description, illustrated above in grey.

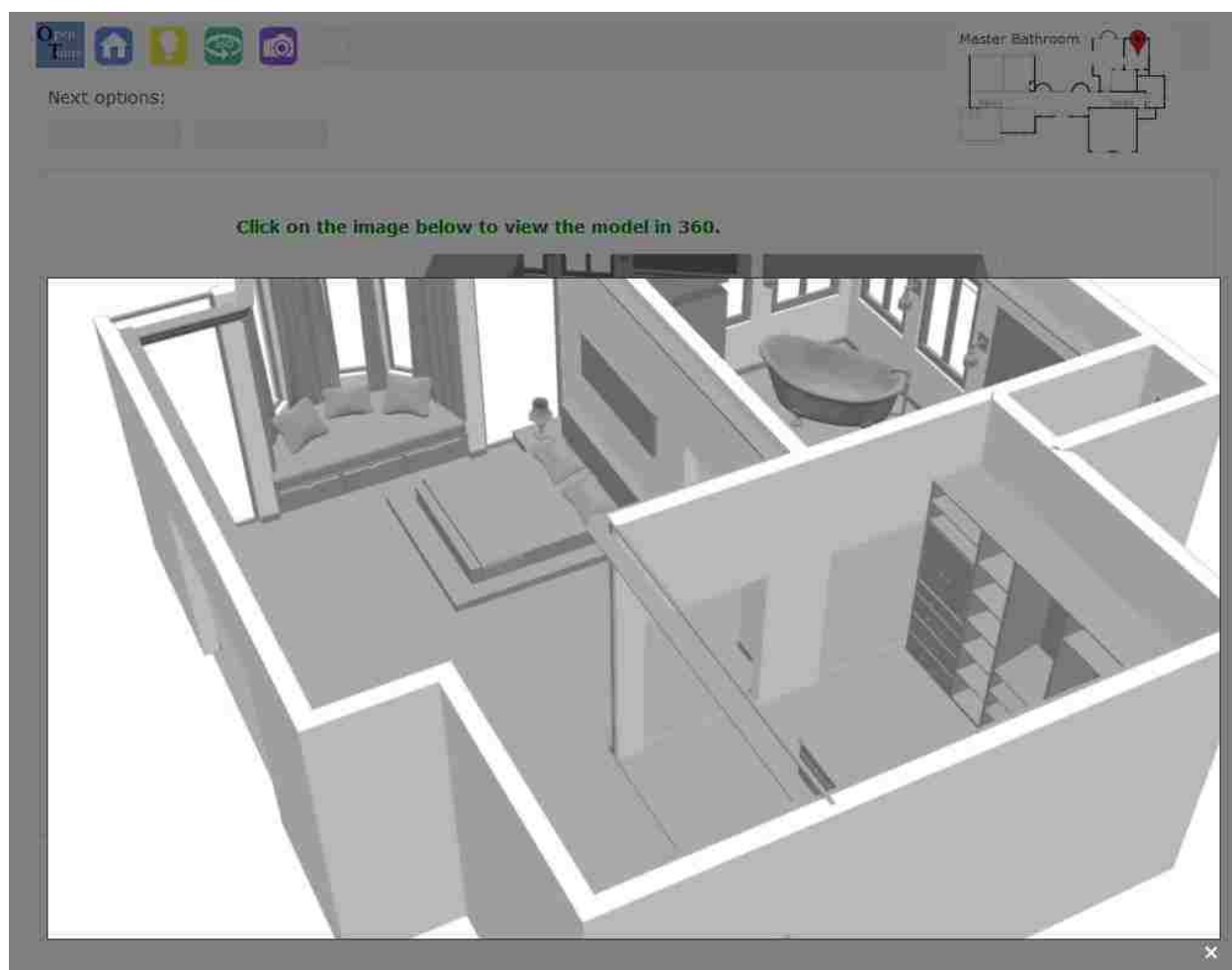


Figure 32: 360 degree rotation screen

3.4.2.2.2 360 degree rotation screen

The purpose of the 360 screen is to provide users with the ability to manipulate a 360 degree view of a specific room. This screen offers an alternative way of viewing 360 degree views compared to some of the options available within virtual tour companies. The information for

the 360 degree view is held under the <threeSixty> tag. This tag holds the “id” information that is used by a PHP file to retrieve a series of sequential images which is then extracted and run on a JQuery 360 rotation script to produce a locked 360 view for the model. Since the view is actually rendered images with a set viewpoint, it prevents accidental rotation of the mouse to view ceilings or floors. This method, although more rigid in control, still has base on the ability to tailor multiple 360 models as need for additional viewpoints. Since each 360 view utilizes only 48 frames of a model, this concept is relatively light weight and can be implemented easily.

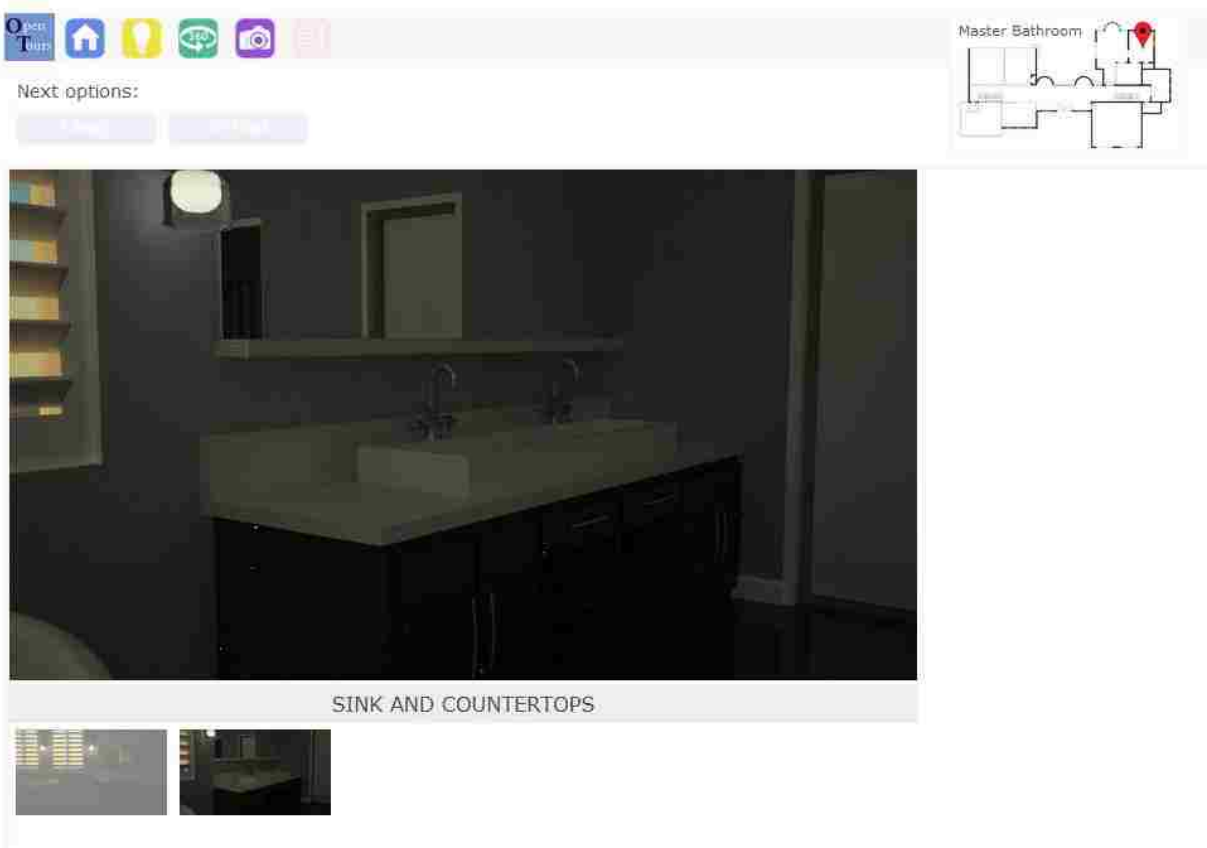


Figure 33: Images Scree

3.4.2.2.3 Image Screen

This image screen is used to provide miscellaneous images produced by the designer that they wish to include for the viewers. This information is held within the <pictures> tag and operates in the same way as the lighting section.

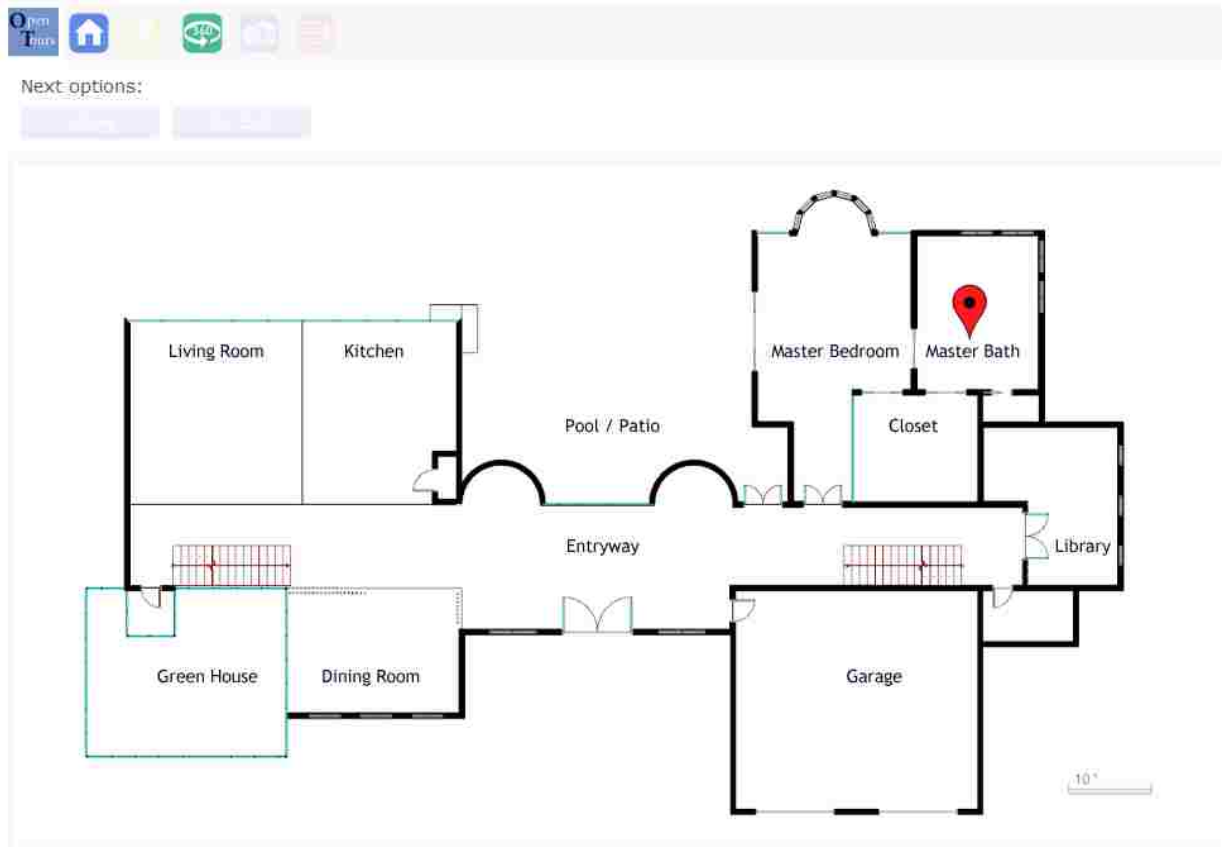


Figure 34: Map page

3.4.2.2.4 Map Screen

The map screen is opened upon clicking on the smaller map present in the other screens in the upper right hand corner. Used to orient the user, the larger screen is meant to hold a more detailed view of the specific location that is not possible in the smaller view. The coordinates of both maps are held within the <map> tag in the xml. The map, or “key plan” information, is generated based on coordinates pertaining to the small map (visible on all pages, except the map page) and the large map (visible on the map page).

3.5 LIMITATIONS / COMPLICATIONS FACED

The development of Open Tours was relatively successful in terms of meeting the prototype goals. There are some remaining challenges faced during production which highlight the limitations of such a tool. One is the amount of time and effort involved in the development and

production of the final product. Another is the struggles and difficulties encountered while using the rendering system efficiently.

3.5.1 *Time and Computing Power*

One important issue is the large amount of time and work which had to be dedicated to the Open Tours project. Aside from the resources necessary in regards to hardware for rendering, which is expendable given that computing will both increase in speed and efficiency in the future, there is the issue of hours a designer has to commit to dedicating in order to complete this process. It took 10 weeks to complete a 3d model, including textures, lighting, animation, and rendering. This amount of time may not be available within industry on particular projects. Therefore development of a solution which could automate parts of this process would reduce the amount of work a designer needs to dedicate.

3.5.2 *Materials and Rendering*

The learning curve to understand the application of materials in respect to global illumination as well as how electrical lights affects a scene was high. The understanding of how each of these aspects affects the length of a given render is additionally complex. One example is the glass like reflection seen in the image below which was generated from a glass material for V-Ray. Note the complexities of the software for the settings on the left in the image below. The options provided are not just numerous, they each require further definition and research in order to be understood properly.

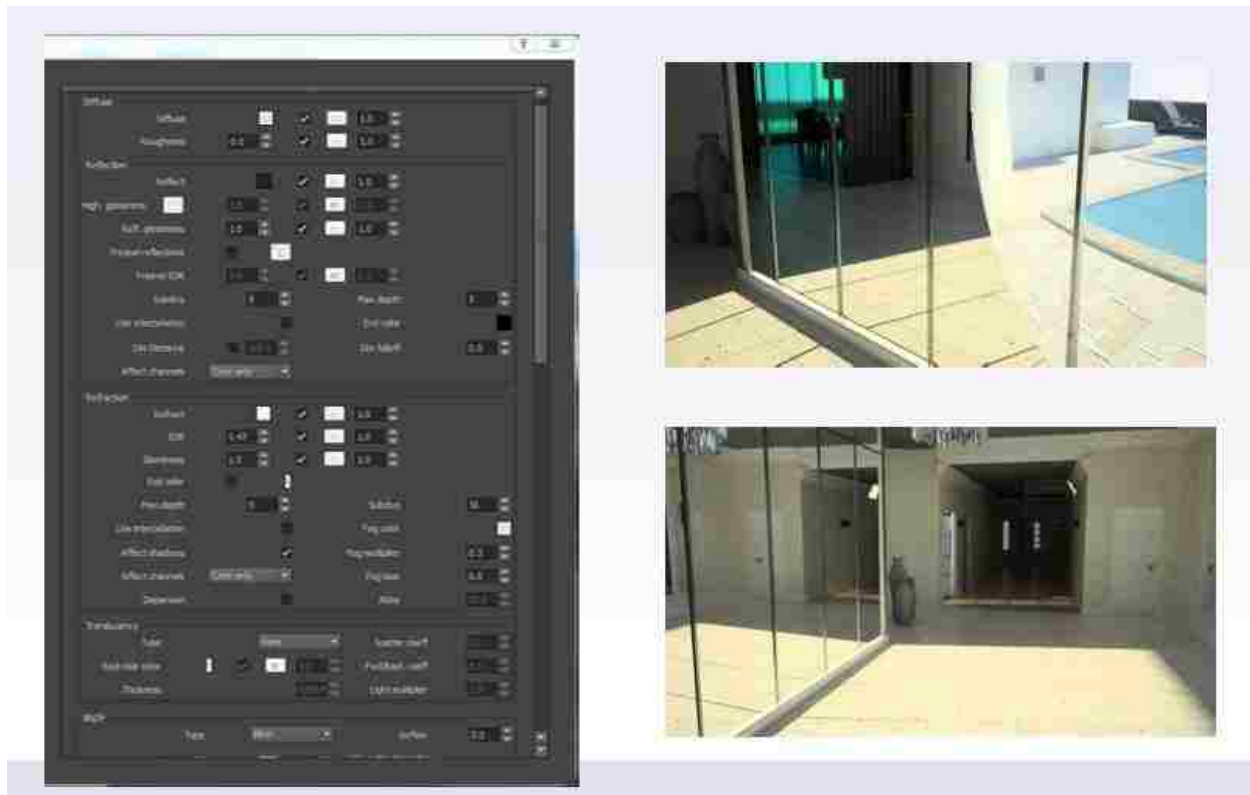


Figure 35: Material settings

A larger frustration though, is that the efforts required to learn and understand these settings was sometimes not related to actual lighting and material calculations. Although the requirement during the renderer selection process did not require a physically based renderer, during the development process based on challenges faced in understanding material and rendering properties, the availability of a standardized singular system which only used attributes available in actual settings rather than ones made available to ease usability for the end user would have been desirable. One, it would provide a single standardized renderer in industrials where the application of real settings mattered. Two, this would allow for a more exact replication of real lighting in the scene. Three, designers would be able to learn a set of information based on actual calculations which could be used towards other elements of lighting rather than an understanding of attributes that are only applicable in a specific renderer with no relevance to real properties. Unfortunately at this time, such standardization is as difficult to

acquire in a renderer as a universal 3d package is within industry. Until this is resolve, designers will be left with balancing out priorities and goals for each available package.

3.6 FUTURE POSSIBILITIES

3.6.1 *Automation*

In order to make Open Tours useful in future development, there needs to be a way to integrate automation into the program, both in respect to camera and animation, as well as coding the XML documents. Since all of these components were completed from scratch, this left a large margin of error and variation as well as time commitment. The idea of automation is possible especially considering the initial set of guidelines written in this thesis. Given a complete set of well-defined guidelines, tailored for 3d animation packages, these setting could be automated for use. If a script was generated to accept inputs for height, linear pace, and rotation, while using locators to mark points in a 3d model to dictate the traveling distance and path, conversion of this information to a camera would be possible. This possibility could even extend to allowing varying angle of view at certain points as well as the inclusion of pause frames in the beginning and end of an animation. The hand coded XML was also subject to error due to the high amount of repetition between code elements. Having the ability to have a computer-generate this information based off a node based room description would eliminate this issue.

3.6.2 *Other XML applications*

Although in Open Tours the XML was used to store node information for spaces, the flexibility and convenience of such as storage system opens up quite a variety of possibilities. XML nodes can be applied to number of other concepts to such as BIM model information, space layout variations, video walk through animation of real estate agencies, and any other concepts that would benefit from a node base system which could utilize information extraction.

3.6.3 *Retention of Information*

Testing of the usability of the Open Tours tool with participations would be an interesting research project. This information is useful since it would indicate how well Open Tours performs in comparison to other applications with users. In addition studies could evaluate the amount of information retained based on this type of visualization compared to other visualization choices such as 2d floor plans or fly thru animations. If the results are positive, this could further allow development of such a tool in the cases of first responders, teaching and education, and other memory allocation exercises.

Chapter 4. CONCLUSION

4.1 CONCLUSION

In conclusion, challenges faced during the development of Open Tours was not necessarily new or elements unfamiliar to the architectural and rendering community. This tool sought to make the decision process with cameras, animations, and renderings a little bit easier for designers, while offering a simpler, more consistent visualization of the information for the users. The completion of this tool opened up some future possibilities in XML data storage, automation, and user cognitive and thought processes as interesting future projects to explore. It is the hopes that Open Tours will provide a positive alternative in viewing animations and renderings for architects and become a stepping stone towards future possibilities using the technologies and components that were implemented.

REFERENCES

- [1] R. Abdullah, M. Christie, G. Schofield, C. Lino and P. Oliver, "Advanced Composition in Virtual Camera Control," *Smart Graphics*, no. Aug 2011, pp. 13-24.
- [2] R. Bekerman, "RonEnbekerman Architectural Visualization Blog," Polytown, 2015. [Online]. Available: <http://www.ronenbekerman.com/photographic-approach-in-architectural-visualisation/>. [Accessed 01 03 2015].
- [3] Cambridge in Colour, "Cambridge in Color," Cambridge in Color, 2015. [Online]. Available: <http://www.cambridgeincolour.com>. [Accessed 15 03 2015].
- [4] Viscorbel, "Viscorbel," 2015. [Online]. Available: <http://viscorbel.com/vray-physical-camera-tutorial/>. [Accessed 01 03 2015].
- [5] Sophia, "Sophia," 2015, [Online]. Available: <http://www.sophia.org/tutorials/design-in-art-balance-and-contrast>. [Accessed 17 03 2015].
- [6] A. Roman, "Third Seventh," The Third and The Seventh, 2015. [Online]. Available: <http://thirdseventh-book.com/contact/>. [Accessed 01 03 2015].
- [7] F. Thomas and O. Johnston, *The Illusion of Life: Disney Animation*, Disney's Hyperion, 1995.
- [8] "Animation," [Online]. Available: <http://wayofthepixel.net/index.php?topic=4355.20>.
- [9] B. Johnson, "Arch 482," 2015. [Online]. Available: <http://courses.washington.edu/arch481/00Homepage/0.default.html>. [Accessed 01 03 2015].
- [10] Radiance, "Radiance Synthetic Imaging System," [Online]. Available: http://radsite.lbl.gov/radiance/refer/Notes/rendering_note.html.
- [11] 3D World, February 2010. [Online]. Available: http://www.thethirdfloorinc.com/publish/_assets/417/10/02/tdw126.f.renderers-7a90.pdf.
- [12] D. Takahashi, "VB News," 24 04 2013. [Online]. Available: <http://venturebeat.com/2013/04/24/the-making-of-pixars-latest-technological-marvel-monsters-university/>. [Accessed 16 05 2015].
- [13] Web3D Consortium, "Web 3D Consortium," 2015. [Online]. Available: <http://www.web3d.org/x3d/what-x3d>. [Accessed 24 05 2015].
- [14] J. Behr, P. Eschler, Y. Jung and M. Zollner, "X3DOM - A DOM-based HTML5 / X3D Integration Model," in *Web 3D*, New York, 2009.
- [15] F. Pittarello and T. Pellegrini, "Designing and experimenting a Web3D application for increasing the environmental awareness," in *ACM*, Trento, 2013.
- [16] Adobe Systems Inc., "Adobe Technical Communication Suite," Jan 2012. [Online]. Available: <http://blogs.adobe.com/tcs/2011/01/framemaker/interactive-3d-models-in-documents.html>. [Accessed 9th Dec 2014].
- [17] Apple, Inc., "QuickTime," 01 06 2009. [Online]. Available: https://developer.apple.com/legacy/library/documentation/QuickTime/InsideQT_QTVR/0Preface/QTVR-preface.html. [Accessed 12 2014].
- [18] C. Calderon, K. Nyman and N. Worley, "The Architectural Cinematographer: Creating Architectural Experiences in 3D Real-time Environments," *International Journal of Architectural Computing*, pp. 71-90, 2006.

- [19] C. Calderon, N. Worley and K. Nyman, "Architectural Cinematographer: An Initial Approach to Experiential Design in Virtual Worlds," in *Computer Aided Architectural Design Futures*, 2005.
- [20] Fourth Dimension Solutions, "Fourth Dimension Solutions," Fourth Dimension Solutions, 1 Jan 2010. [Online]. Available: <http://www.fourthdimensionsolutions.com>. [Accessed 10 03 2015].
- [21] TourDe Force 360VR, "TourDeForce 360," Tour De Force , 2015. [Online]. Available: <http://www.tourdeforce360.com>. [Accessed 1 03 2015].
- [22] Village Features LLC., "Village Features," Village Features, 2015. [Online]. Available: <http://villagefeatures3d.com>. [Accessed 01 03 2015].
- [23] Yantram Studio, "Yantram Studio," Yantram Studio, 2015. [Online]. Available: <http://www.yantramstudio.com>. [Accessed 01 03 2015].
- [24] C.-L. Chen and B. R. Johnson, "DVIN - a Dual-View Information Navigator," in *ACADIA* , 2007.
- [25] M. Stepp, J. Miller and V. Kirst, "Web Programming, Step by Step," Marty Stepp, 2011.
- [26] K. McHenry and P. Bajcsy, "An Overview of 3D Data Content, File Formats, and Views," Image Spatial Data Analysis Group, Urbana, IL, 2008.
- [27] J. Lasseter, "Tricks to Animating Characters with a Computer," 1994. [Online]. Available: https://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/principles/lasseter_s94.htm.
- [28] C. G. "Chaos Group, Innovative Rendering Technologies," Chaos Group, 2014. [Online]. Available: http://help.chaosgroup.com/vray/help/150SP1/tutorials_imap2.htm.
- [29] C. Lino, "The director's lens: an intelligent assistant for virtual cinematography," in *MM*, 2011.
- [30] H. W. Jensen, *Realistic Image Synthesis Using Photon Mapping*, 2001.
- [31] E. Carnielli and F. Pittarello, "Interactive stories on the net: a model and an architecture for X3D worlds," in *Web3D*, New York, 2009.
- [32] A. Preetham, P. Shirley and B. Smits, "A Practical analytic model for daylight," in *Siggraphy*, New York, 1999.

BIBLIOGRAPHY

- 3D World. (2010, February). Retrieved from http://www.thethirdfloorinc.com/publish/_assets/417/10/02/tdw126.f.renderers-7a90.pdf
- Abdullah, R., Christie, M., Schofield, G., Lino, C., & Oliver, P. (n.d.). Advanced Composition in Virtual Camera Control. *Smart Graphics*(Aug 2011), 13-24.
- Adobe Systems Inc. (2012, Jan). Adobe Technical Communication Suite. Retrieved Dec 9th, 2014, from <http://blogs.adobe.com/tcs/2011/01/framemaker/interactive-3d-models-in-documents.html>
- Animation. (n.d.). Retrieved from <http://wayofthepixel.net/index.php?topic=4355.20>
- Apple, Inc. (2009, 06 01). QuickTime. Retrieved 12 2014, from https://developer.apple.com/legacy/library/documentation/QuickTime/InsideQT_QTVR/0Preface/QTVR-preface.html
- Behr, J., Eschler, P., Jung, Y., & Zollner, M. (2009). X3DOM - A DOM-based HTML5 / X3D Integration Model. Web 3D. New York.
- Bekerman, R. (2015). RonEnbekerman Architectural Visualization Blog. (Polytown) Retrieved 03 01, 2015, from <http://www.ronenbekerman.com/photographic-approach-in-architectural-visualisation/>
- Calderon, C., Nyman, K., & Worley, N. (2006). The Architectural Cinematographer: Creating Architectural Experiences in 3D Real-time Environments. *International Journal of Architectural Computing*, 71-90.
- Calderon, C., Worley, N., & Nyman, K. (2005). Architectural Cinematographer: An Initial Approach to Experiential Design in Virtual Worlds. *Computer Aided Architectural Design Futures*.
- Cambridge in Colour. (2015). Cambridge in Color. (Cambridge in Color) Retrieved 03 15, 2015, from <http://www.cambridgeincolour.com>
- Carnielli, E., & Pittarello, F. (2009). Interactive stories on the net: a model and an architecture for X3D worlds. Web3D. New York.
- Chaos Group. (2014). Chaos Group, Innovative Rendering Technologies. (Chaos Group) Retrieved from http://help.chaosgroup.com/vray/help/150SP1/tutorials_imap2.htm
- Chen, C.-L., & Johnson, B. R. (2007). DVIN - a Dual-View Information Navigator. *ACADIA*.
- Fourth Dimension Solutions. (2010, Jan 1). Fourth Dimension Solutions. (Fourth Dimension Solutions) Retrieved 03 10, 2015, from <http://www.fourthdimensionsolutions.com>
- Jensen, H. W. (2001). Realistic Image Synthesis Using Photon Mapping.
- Johnson, B. (2015). Arch 482. Retrieved 03 01, 2015, from <http://courses.washington.edu/arch481/00Homepage/0.default.html>
- Lasseter, J. (1994). Tricks to Animating Characters with a Computer. Retrieved from https://www.siggraph.org/education/materials/HyperGraph/animation/character_animation/principles/lasseter_s94.htm
- Lino, C. (2011). The director's lens: an intelligent assistant for virtual cinematography. *MM*, (pp. 323-332).
- McHenry, K., & Bajcsy, P. (2008). An Overview of 3D Data Content, File Formats, and Views. Urbana, IL: Image Spatial Data Analysis Group.
- Pittarello, F., & Pellegrini, T. (2013). Designing and experimenting a Web3D application for increasing the environmental awareness. *ACM*. Trento.
- Preetham, A., Shirley, P., & Smits, B. (1999). A Practical analytic model for daylight. *Siggraphy*. New York.
- Radiance. (n.d.). Radiance Synthetic Imaging System. Retrieved from http://radsite.lbl.gov/radiance/refer/Notes/rendering_note.html
- Roman, A. (2015). Third Seventh. (The Third and The Seventh) Retrieved 03 01, 2015, from <http://thirdseventh-book.com/contact/>

- Sophia. (n.d.). Sophia. (2015) Retrieved 03 17, 2015, from <http://www.sophia.org/tutorials/design-in-art-balance-and-contrast>
- Stepp, M., Miller, J., & Kirst, V. (2011). Web Programming, Step by Step. Marty Stepp.
- Takahashi, D. (2013, 04 24). VB News. Retrieved 05 16, 2015, from <http://venturebeat.com/2013/04/24/the-making-of-pixars-latest-technological-marvel-monsters-university/>
- Thomas, F., & Johnston, O. (1995). The Illusion of Life: Disney Animation. Disney's Hyperion.
- TourDe Force 360VR. (2015). TourDeForce 360. (Tour De Force) Retrieved 03 1, 2015, from <http://www.tourdeforce360.com>
- Village Features LLC. (2015). Village Features. (Village Features) Retrieved 03 01, 2015, from <http://villagefeatures3d.com>
- Viscorbel. (2015). Viscorbel. Retrieved 03 01, 2015, from <http://viscorbel.com/vray-physical-camera-tutorial/>
- Web3D Consortium. (2015). Web 3D Consortium. Retrieved 05 24, 2015, from <http://www.web3d.org/x3d/what-x3d>
- Yantram Studio. (2015). Yantram Studio. (Yantram Studio) Retrieved 03 01, 2015, from <http://www.yantramstudio.com>