THE ART OF PREPRODUCTION: FROM CONCEPT TO COMPUTER

A Project
Presented
to the Faculty of
California State University, Chico

In Partial Fulfillment
of the Requirements for the Degree
Master of Arts
in
Interdisciplinary Studies
Applied Computer Graphics

by
Lindsey Erin Anderson
Spring 2010
THE ART OF PREPRODUCTION: FROM CONCEPT TO COMPUTER

A Project

by

Lindsey Erin Anderson

Spring 2010

APPROVED BY THE INTERIM DEAN OF THE SCHOOL OF GRADUATE, INTERNATIONAL, AND INTERDISCIPLINARY STUDIES:

_________________________________
Mark J. Morlock, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Sara A. Trechter, Ph.D.
Graduate Coordinator

Clarke H. Steinback, Ph. D., Chair

Susan L. Whitmore, M.F.A.

Frank Armstrong, M.F.A.
ACKNOWLEDGMENTS

I’d like to take this opportunity to thank those whom have supported me throughout the entirety of this project. First, my parents who love me very much, and encouraged me to pursue my education for as long as I felt I needed to. Second my friends who have been both my greatest source of strength and inspiration. And finally to the students and faculty of the APCG program, you are all amazing.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vi</td>
</tr>
<tr>
<td>Abstract</td>
<td>vii</td>
</tr>
<tr>
<td>CHAPTER</td>
<td></td>
</tr>
<tr>
<td>I. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Purpose of the Project</td>
<td>1</td>
</tr>
<tr>
<td>Limitations</td>
<td>4</td>
</tr>
<tr>
<td>II. Literary Review</td>
<td>6</td>
</tr>
<tr>
<td>Game History</td>
<td>6</td>
</tr>
<tr>
<td>Current Generation of Consoles</td>
<td>8</td>
</tr>
<tr>
<td>Methods of Texturing</td>
<td>9</td>
</tr>
<tr>
<td>Tiling Textures</td>
<td>9</td>
</tr>
<tr>
<td>UV Textures</td>
<td>11</td>
</tr>
<tr>
<td>Effect Maps</td>
<td>17</td>
</tr>
<tr>
<td>Texturing Tools</td>
<td>20</td>
</tr>
<tr>
<td>Modeling Tools</td>
<td>21</td>
</tr>
<tr>
<td>Game Engines</td>
<td>23</td>
</tr>
<tr>
<td>Summary</td>
<td>25</td>
</tr>
<tr>
<td>III. Analysis of Texturing Tests</td>
<td>26</td>
</tr>
<tr>
<td>Introduction</td>
<td>26</td>
</tr>
<tr>
<td>Development of Texturing Tests</td>
<td>26</td>
</tr>
<tr>
<td>The Survey Instrument</td>
<td>33</td>
</tr>
<tr>
<td>Survey Results Analysis</td>
<td>35</td>
</tr>
<tr>
<td>Intended Size Texture vs. Scaled Texture Tests</td>
<td>36</td>
</tr>
<tr>
<td>Tiling Texture Tests</td>
<td>38</td>
</tr>
<tr>
<td>UV Texturing Samples l</td>
<td>41</td>
</tr>
</tbody>
</table>
## Chapter 5: Recommendations and Conclusions

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>67</td>
</tr>
<tr>
<td>Limitations to the Findings</td>
<td>70</td>
</tr>
<tr>
<td>Recommendations</td>
<td>71</td>
</tr>
<tr>
<td>General Guidelines to Texturing for Video Games</td>
<td>72</td>
</tr>
<tr>
<td>Non-Destructive Photoshop Texturing Practices</td>
<td>72</td>
</tr>
<tr>
<td>Archiving Textures</td>
<td>74</td>
</tr>
<tr>
<td>Findings from Texturing Tests</td>
<td>75</td>
</tr>
<tr>
<td>Conclusion</td>
<td>76</td>
</tr>
</tbody>
</table>

## References

<table>
<thead>
<tr>
<th>References</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>77</td>
</tr>
</tbody>
</table>

## Appendices

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.</td>
<td>Storyboard</td>
<td>81</td>
</tr>
<tr>
<td>B.</td>
<td>Model and Texture List</td>
<td>133</td>
</tr>
<tr>
<td>C.</td>
<td>Survey Examples</td>
<td>136</td>
</tr>
<tr>
<td>D.</td>
<td>Wood Tutorial</td>
<td>145</td>
</tr>
<tr>
<td>E.</td>
<td>Rust Tutorial</td>
<td>176</td>
</tr>
<tr>
<td>F.</td>
<td>Concrete Tutorial</td>
<td>190</td>
</tr>
<tr>
<td>G.</td>
<td>Normal Map Tutorial</td>
<td>217</td>
</tr>
<tr>
<td>H.</td>
<td>In-Game Screenshots</td>
<td>228</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10</td>
</tr>
<tr>
<td>2.</td>
<td>11</td>
</tr>
<tr>
<td>3.</td>
<td>12</td>
</tr>
<tr>
<td>4.</td>
<td>12</td>
</tr>
<tr>
<td>5.</td>
<td>13</td>
</tr>
<tr>
<td>6.</td>
<td>28</td>
</tr>
<tr>
<td>7.</td>
<td>30</td>
</tr>
<tr>
<td>8.</td>
<td>31</td>
</tr>
<tr>
<td>9.</td>
<td>31</td>
</tr>
<tr>
<td>10.</td>
<td>32</td>
</tr>
<tr>
<td>11.</td>
<td>37</td>
</tr>
<tr>
<td>12.</td>
<td>40</td>
</tr>
<tr>
<td>13.</td>
<td>42</td>
</tr>
<tr>
<td>14.</td>
<td>44</td>
</tr>
<tr>
<td>15.</td>
<td>47</td>
</tr>
<tr>
<td>16.</td>
<td>62</td>
</tr>
</tbody>
</table>
ABSTRACT

THE ART OF PREPRODUCTION: FROM CONCEPT TO COMPUTER

by

Lindsey Erin Anderson

Master of Arts in Interdisciplinary Studies

Applied Computer Graphics

California State University, Chico

Spring 2010

The creation of textures is a vital part of video game creation. For those who wish to learn how to make textures there exists little information available that is not contradictory or incomplete. The goal of this project was to provide practical information to those getting started in the field of texturing.

The primary component of this project was the creation and comparison of a series of texturing procedures, developed to determine methods of generating high quality in-game assets in the most efficient ways possible. The products of these procedures were used in a survey administered to the public as a means of gauging the quality of the textures produced. Based on the survey results and personal observation, a digital environment was made showcasing what was learned from the texturing tests.
Appendices at the end of the project give step-by-step instruction on the creation of the common types of textures and the creation of normal maps.

This project is not an absolute authority on creating textures for video games, but should be taken as a series of guidelines or suggestions. Test results, personal observations and the description of the creation of the digital environment are intended help the reader make informed decisions on the creation of content and how to start texturing.
CHAPTER I

INTRODUCTION

The creation of textures is a vital part of video game production. Textures are important because they quickly communicate information to the player while adding color and depth to a game world. For those who wish to learn how to make effective textures, the majority of information available is either contradictory or incomplete. My project was designed to explore the contradictory information and pull from it the techniques that could be used to consistently create high quality textures. This was done by creating a series of texture samples that compared the outcomes of said techniques. The texture samples produced were used in a survey presented to participants online. The purpose of the survey was to gather feedback rating the texture samples on visual quality. Based on survey results, a small digital environment was created as a means of showcasing texturing processes learned and developed throughout the course of the project. Techniques developed in this project are meant to be easily replicated by others, particularly those learning to texture for the first time.

Purpose of Project

The purpose of this project was to answer basic questions about the process of texturing for video games. When I began researching the subject of texturing I had many questions for which I did not find clear answers. Questions I wanted to answer were: if it
was acceptable to create large textures then reduce them to their intended size, if players
could recognize a difference in texture sizes, if there was a noticeable difference in
texture quality between tiled textures or UV textures and which programs created the
highest quality normal maps with the least amount of user familiarity. I wanted to
examine resizing textures because some of the sources researched stated that it was
acceptable to create a texture at double its needed size then scale it down. Other sources
discouraged this practice, stating that resizing would always result in image distortion.
There did not seem to be a straight yes or no answer to this question. I questioned if
players saw the difference between large and small sized textures when playing a game
because during my research I found user created game modifications that doubled or
tripled the size of textures in-game. The creators of the modifications did this because
they believed that the larger textures looked better. I wanted to compare which texturing
method tiling textures or UV mapped textures produced higher quality images because
they are two very different ways of applying texture to an object. Tiling textures are small
images repeated indefinitely over a given amount of space. UV textures are unique to the
model they to which they are applied. Theoretically, tiled textures would be faster to
create and apply, and would give better visual results since there is no wasted space in the
texture. UV textures, which are slower to produce, always lose some detail due to the
nature of their creation. However, these textures are highly desirable because they are
unique to the model so add variety to an environment. Lastly, I wanted to look at the
creation of normal maps in different available programs to see which gave the best
normal map results with the least amount of familiarity to that program to find which
would give me the best results fastest.
Textures

A texture is a two dimensional image applied to a digital three-dimensional object. A successful texture mimics real world texture to contribute to viewer engagement. This project would focus on creating textures by hand without the use of photographs in a digital medium.

Normal Maps and Parallax Mapping

Normal maps and parallax mapping uses an image to simulate surface details without using complex geometry which frees up a computer’s processor for other tasks. Texture artists developing for new games need to know how to create effective normal maps because some engines, like the one used in this project, will not display textures correctly without a normal map. Normal map samples would be created in four different programs to determine which one produced the most effective maps.

Digital Game Engines

To make a quick distinction, game engines are specialized software that consolidate processes, such as rendering, physics, lighting and other functions, which together make up a game. Take away the engine and the game will no longer run. Since this project focuses on textures, which are images imported into a game engine, the way in which an engine handled textures was important. This engine would be used for both the texturing tests and digital environment. Several engines were considered before this project started, including Unreal 2 (Epic Games, 2008), Half-Life 2: Source (Valve, 2007), the Neverwinter Nights 2 Electron tool kit (Atari, n.d.) and The Elder Scrolls: Construction Set 4 from The Elder Scrolls IV: Oblivion (Bethesda Softworks, 2009).
The Survey

A survey was developed to gather feedback from other people about their observations of my texture samples. Feedback was important because it would help determine which methods of producing textures yielded the highest quality results. The survey was also important because it gave me something to compare to my own observations of the texture samples. It asked participants, both gamers and non-gamers, to look at texture samples and rate them according to establish criteria for what constituted a successful texture.

Adaptation of Research and Implementation:
Game Modification

In order to apply the results of the texturing survey, a small digital environment was created and textured using techniques developed during the course of the project. This one-person project included storyboards, models, and textures and a digital environment created using The Elder Scrolls: Oblivion (TES) and its editor TES: Construction Set 4.

Limitations

The scope of this project needed to be kept to a manageable size for a one-person team. Restrictions were placed on the generation of texture samples and choice of software packages. Three sizes of textures were used to create samples; 512x512 pixels, 1024x1024 pixels and 2048x2048 pixels. These sizes refer to the actual size of the texture. A 17” monitor set to 1280x1024 pixels was used to retain a consistent texture to pixel ratio though out the duration of the project. Four programs were used to make normal maps: ZBrush v.2, Mudbox v.2, Silo v.2, and Photoshop CS2 with the NVIDIA
filter plug-in. Photoshop was used to generate textures. The game used for displaying texture samples and the digital environment was Bethesda Softwork’s *The Elder Scrolls: Oblivion*, edited with TES: Construction Set 4. The digital environment was not a full conversion modification. Full conversions require replacing every model, texture and animation in the game with custom made content to completely change the game. This project does not include an in-depth discussion on the 3D modeling process, does contain a brief discussion about how models influenced the creation of the textures.

Chapter II will discuss some of the history behind video games to show how far the industry has progressed and the dramatic leaps made in game technology in the past forty years. It also discusses different texturing techniques, books, websites, engines, and software researched in preparation for the project. Chapter III explores the creation of texture samples and results of the survey. Feedback from the public and my opinions of the texture samples will be examined in this chapter as well as how this information impacted the production of the digital environment. Chapter IV deals with the creation of the example digital environment, from planning phase to end phase. Chapter V contains a summary of findings, recommendations for new texture artists and final conclusions.
CHAPTER II

LITERARY REVIEW

In forty years, video games have undergone a dramatic evolution, beginning with simple lines and dots then growing rapidly into graphical extravaganzas produced by a multi-million dollar industry. Looking back, we can see a steady progression towards more sophisticated hardware and software. New generations of console and computer games have steadily jumped in graphical capability as digital games became more popular and more profitable. Advertising the graphical capabilities of consoles has been a quick way for developers to attract customers. The jumps in graphic capabilities have pushed back memory limitations, increasing the amount detail of detail a game can display onscreen. For game makers, better technology means better tools to work with, resulting in better looking games. However, it also means that artists have to quickly adapt to new tools and develop techniques to leverage new technologies.

Game History

The first games were black and white and extremely simple graphically, consisting solely of lines and boxes. Due to memory constraints, graphics were vector based, which meant no bitmapped graphics and therefore no textures. Visuals were created by the same programmers who coded the games, with typically no more than one or two people working on a game (Kent, 2001).
The 1980s saw the 8-bit systems such as the Nintendo Entertainment System (NES). The graphics of this era were blocky and crude, but unlike previous generations, characters and environments could be fashioned into recognizable shapes. In these games, we see some of the earliest instances of tiling graphical elements. For example, in *Super Mario Brothers*, the main character Mario runs upon the same brick throughout an entire level. While bricks like these were not truly textures as we know them, reusing graphical elements is a practice that has continued to this day. The late 1980s to the early 90’s saw a doubling in processor power, and the ushering in of the 16-bit era, where consoles like the Super Nintendo Entertainment System and Sega’s Genesis could display hundreds of colors.

The next landmark release of consoles saw machines that were capable of processing simple textured 3D models in real time. Now, in addition to artists who could draw, animate and paint there existed a need for 3D modelers and texture artists. In 1993, three games were released that used 3D graphics in new, interesting ways: *Myst*, *The 7th Guest*, and *Doom*. *Myst* garnered critical acclaim with its pre-rendered backgrounds, puzzles and storyline (Cyan Worlds, n.d.). *Myst* was not a true 3D game at the time of its release. Instead, the entire game is a slideshow of pre-rendered images with hotspots where puzzles or animation would be (Hutchison, 2008). *The 7th Guest* was also a pre-rendered puzzle game, but unlike *Myst*, *The 7th Guest* took a horror approach (Mobygames, 2009a). *Doom* kicked off the popularity of the FPS (first person shooter) genre; where players were surrounded by a game world they could freely move though in three dimensions, looking through the eyes of their character. The game kept players on their toes with hordes of monsters, fast-paced action, and the ability to play with other
people over a network (Kent, 2001). These games caught the public’s attention with higher levels of visual detail than had ever previously been seen thanks to the inclusion of 3D elements.

Though there was the possibility for more detailed visuals in games, there were still technical limits which needed to be observed. Models had to be limited to a handful of polygons. Too many polygons on screen could overload a machine’s memory with calculations, causing it to run either slowly or not at all. Textures had to be small in file size because they potentially took up more room than the models or the game itself, causing some pre-rendered and real time games to span five or more CDs. The rule of thumb became to reserve high-density models for more important elements: objects that would be seen up close or characters. Other models would be made with as few polygons as possible, relying on small textures to make up the detail.

Current Generation of Consoles

Presently, the visuals of games are nearing photorealistic quality. Home computers and game consoles are commonly shipped with multiple core processors, capable of crunching more numbers than ever before. High-end gaming computers are capable of using multiple GPUs, making real-time rendering of high polygon count models and large textures possible. New game engines easily support the application of multiple image maps to a single object. Map types range from color maps to special effects maps that alter the look of an object. These maps are explained later in the chapter.
Art teams for games are large with several sub-departments within the art department umbrella. Today, an art department can consist of any combination of animators, prop modelers and texture artists, background artists, environmental modelers and texture artists, character modelers and texture artists, cinematic artists, and storyboard artists. Current game engines are capable of high levels of detail, forcing artists to find new ways of making the most of the power available to them. Texture artists must learn to work with larger image maps which allow them to easily integrate photographic materials into their textures without losing image quality (Linde, 2004). Since engines now allow multiple textures to be applied to objects, the workload of the texture artist has increased dramatically. Techniques to reuse texture assets are encouraged, but at the same time, the sheer number of assets required forces the artist to find efficient ways to create as many high quality texture assets within the shortest amount of time as they can.

Methods of Texturing

A texture artist must be intimately familiar with the tools and techniques at their disposal. For the purposes of this project, techniques used to make textures will be discussed first, followed by descriptions of which tools were utilized for this project.

Tiling Textures

Tiling textures are similar to home decorating wallpaper (see Figure 1 for an example). They are small images repeated over a large area with no noticeable breaks in the image; such breaks are called seams. However, unlike real wallpaper, repeating patterns are undesirable in digital games. A tiled texture needs to stay uniform across the
Figure 1. A tiling texture.

entirety of the image; no one part of the texture can stand out more than another. Unintended patterns can develop when a tiling texture has noticeable irregularities. These patterns destroy the believability of an object or environment, breaking the player’s immersion in a game.

In spite of these challenges, tiling textures do have one key advantage: they allow developers to fill large amounts of space with one small, memory-efficient texture (as seen in Figure 2). Tiled textures can also be applied to multiple surfaces, since the texture is not specific to any one object. Iterating a texture across multiple surfaces or objects has benefits. As an example, using a single wall texture on a village of huts gives the environment visual cohesion, as if it had been purposely built that way by its inhabitants. A tiling texture can be applied to a model either in a 3D modeling program or within a game engine editor. However, tiling textures have limitations. They are
generalized and cannot be used to texture objects like characters. Additionally, if tiling textures are not used imaginatively they can lead to a monotonous environment.

**UV Textures**

The more complicated process of UV texturing creates a texture for a single object. This texture is unique to the model it is made for, unlike a tiled texture that can be applied to many different models. An analogy for the process is to think of a digital model as if it were an orange. Remove the peel of the orange, flatten it out, paint on the peel, and put the painted peel back around the fruit. The peel is the UV map (as seen in Figure 3) and texture (as seen in Figure 4), the fruit is the digital model. A model has both a UV map and a UV texture; a UV map is not a UV texture.

A UV texture takes longer to produce than a tiled texture because a UV map must be laid out before texture creation can begin. The more complicated the object, the
Figure 3. Example of a UV map.

Figure 4. Example of a UV texture.
more complicated the UV map will be, increasing the amount of time it takes to lay out. In 3D modeling programs, models are built on X, Y and Z axis: width, depth and height. A UV map is the 2D representation of the 3D axis. The U and V refer to the coordinates of the UV map, U is up and down, V is left and right. Peeling and flattening out the skin of the model is sometimes more time consuming than painting the texture because one must take into account where breaks in the polygons are. These breaks, also called seams, are eliminated though creative texturing and careful placement of UV map elements. Thoughtfully laid out maps are time well spent, as they are much easier to apply textures to (van der Byl, 2003). UV mapping works best with unique models; a character, special objects, etc. (see Figure 5 for an example).

Figure 5. Example of UV texture applied to a model.
Digital Painting

Digital painting is executed with a computer running image-editing software. Painting is achieved though input with either a mouse or a graphics tablet. Painting with a mouse is physically strenuous and not recommended. Graphics tablets are input devices that consist of a tablet-shaped peripheral plugged into a computer and a pen synchronized to the tablet. With a tablet, the artist can draw or paint much as they might with pen and paper. Working with a tablet does create a disconnection between the artists’ hands and eyes. The artist must train himself/herself to look at a screen while not watching his/her hand, much the way blind gesture drawing is done in art; not looking at one’s hand while drawing may sound easy, but is actually a significant hurdle for the artist to overcome. Compounding the experience is lack of feedback from the tablet pen. The pen, despite the name, does not write in the real world. Even if one did look down while drawing on a tablet, there would be nothing there to orient oneself to. Once an artist is able to overcome these hurdles, painting digitally is a great time saving advantage. Painting textures is an easy way to apply texture to a model. There is less risk of pixilation and when painting UV textures the image is easily fitted to the model. However if one does not have a background in traditional art media it makes mastering the skill much more difficult. Creating a texture that actually tiles seamlessly can be difficult at first. One must learn how to make images that are interesting, but at the same time do not have any one part of the image start out more than another.

Creating a tiled texture is done by digitally painting an entire canvas, then shifting it so that what were once the edges are moved to the middle of the texture. The artist then removes those edges, which causes the texture to tile without seams.
When painting UV textures, “painting inside the lines” becomes a necessary habit. The lines of a UV map represent the polygons of a model (see Figures 3 and 4). Anything painted outside those lines will not be visible on a model.

**Photographs**

Using photographs can be a speedy way of generating textures. What may take hours for a texture painter to emulate with digital paint may take another half the time with photographs and look more realistic. Additionally, photographs are beneficial to those who do not have a background in traditional art media and would not be able to paint a suitable facsimile in a reasonable amount of time. Conversely, those with a background in art can photograph textures made with traditional art media in order to process them on a computer. Such photos still need to be processed into a usable texture. For tiling textures that means cropping a photograph to a usable size, editing the texture so it tiles, and minimizing any distracting patterns. The texture artist must do this while trying to prevent edited portions of the image from becoming blurry.

Using photographs to create UV textures is another option available to the texture artist. Especially if one is photographing traditional art media UV textures (Demers & Urszenyi, 2002). The same issues involved with making tiling textures apply here. When using photographs to create UV textures one must be cautious. In order to fit a photograph to a UV map, it must be edited, this involves operations such as: shrinking an image, rotating it, stretching it, or even flipping it. These editing operations can cause blurriness because they require the editing program to recalculate the positions of pixels in the image. Blurriness is unwanted because it drags the visual quality of a texture down.
Scanning

Scanning textures involves the same problems and benefits as photographing textures. More editing is always required to create a tiling texture or to create a UV. Over editing the scanned image can degrade it, causing blurriness. Yet there are advantages to scanning over photography. Materials such as paper, fabric and other flat media are easily acquired with less chance of light contamination. Scanned images can also be of higher quality than photographs, some scanners going up as high as 1200 dpi.

Resource Comparisons

Only a handful of the sources researched had much to say about creating tiling textures other than using them as a way of filling walls. Each source described different methods of creating tiling textures, if they made any mention. Some worked solely from photographs (Franson, 2004), some worked from scratch (van der Byl, 2003; Aheam, 2006; Mathis, n.d.), still others used traditional art media as part of their process (Demers & Urszenyi, 2002). All of the sources used Photoshop, but what was rarely discussed was how a Photoshop file should be set up. One source said a texture sized 512x512 pixels displays enough detail and that textures sized 1024x1024 pixels was more detail than a player could perceive (Franson, 2004). The rest of the sources recommended testing first, but keeping sizes near 512x512 pixels. Nearly all said that resizing was acceptable, except one who asserted that resizing would always create distortion in a texture (Mathis, n.d.). None discussed whether a tiled texture would look better in place of a UV texture. Only one source discussed using different texture sizes for objects of higher or lower priority (Linde, 2004).
Effect Maps

Effect maps are additional images used to affect the look and feel of an object in a game engine. Now these maps are supported by on-the-fly rendering game engines, further increasing possible detail in-game. There are several different types of maps available to texture artists, but the ones used for this project will be discussed.

Normal Maps and Parallax Mapping

Normal maps have become industry standard for enhancing both textures and models. For the engine used in this project, normal maps were mandatory. If a normal map was not associated with a model it would be displayed as bright purple or black in-game; this may not be the case for other game engines. Normal maps are a way of cutting down polygon counts while retaining a high level of detail. Much like a bump map, normal maps can make the surface of an object look rough, smooth, worn, or pitted.

Bump mapping is a process of creating simulated height and depth with a 2D grayscale image. Black is the absolute deepest shadow, white the absolute highest highlight and middle gray the base. When the model is rendered the areas where the image was dark look pushed down, while the places where the image was light look pushed up. The model is not physically changed in any way, but the illusion is convincing. Normal maps simulate height and depth as well, but use an RGB image (red, green, blue) as opposed to black and white. Using two channels (red and green), as opposed to one (grayscale) gives more accurate detail because the engine is receiving more information to work with (Cloward, n.d.).

Like any other process, there are several ways to create normal maps. One way is to make two versions of a model, one with a low polygon count and another with a
high polygon count. The high count model may be somewhere around a million or more polygons, completely unpractical for a game. However, creating a normal map from the high count model and applying it to the low count model gives the illusion that the low count model looks like the high count one. Add on textures which emphasize the normal map and a low polygon model very similar to its high polygon counterpart is ready to use. This process is suited to working with unique models, such as characters, since the resulting normal map is tailor made for a single model. Another way to create a normal map is to “paint” it in a hybrid program, such as ZBrush. Yet another way to create a normal map is in Photoshop with NVIDIA’s normal map filter (Linde, 2004).

In addition to normal mapping, there is also parallax mapping, which makes use of the third blue channel. Parallax and normal maps are made the same way, but some game engines do not make use of the third blue channel. TES: Oblivion, the game used for this project, uses parallax maps; therefore, it 'sees' the blue channel (KatsBits, 2008). Parallax mapping is a way of seeing simulated depth from different angles. If one looks at a normal map from the side, instead of straight on, the normal map looks flat. With parallax mapping, the blue channel dictates depth from any viewing angle, so where a normal map would look flat parallax mapping looks three dimensional.

Specularity Maps

Specularity maps are grayscale images that dictate how shiny surfaces are. White represents the shiniest parts, black the dullest and middle gray the base. Defining surface specularity lends believability to a setting. Viewers unconsciously expect surfaces shine in a certain way. Weathered wood looking shiny is out of place with what the viewer knows of the real world, weathered wood should be dull, while oiled wood should
be shiny (Linde, 2004). When that expectation is not met in a sufficient manner, it breaks immersion within a game world.

**Alpha Maps**

When an image is loaded into a program such as Photoshop it is comprised of three channels; red, blue and green. These channels record the color information of an image, but there is a forth channel as well, an alpha channel. Alpha channels add transparency to textures. For example, say one wanted to create a rectangular rug with tassels. Modeling tassels would add hundreds of unnecessary polygons, but would lend a heightened level of interest to the model itself. Instead of modeling tassels, a single polygon could be used for the whole rug if the texture has an alpha map applied to it. Negative space around the tassels is removed from an image of a rug, leaving the area transparent. The rug image is saved with the extra channel and then applied to the model. When the model goes into the engine, the negative space around the tassels is transparent; making it look like the polygon is cut to the shape of the tassels. Hours spent modeling tassels are saved and the computers’ processor is freed of calculating unnecessary polygons. Thus, more rugs can be used in the environment, increasing the overall detail of the world (Linde, 2004).

**MIP Maps**

MIP maps are progressively smaller, less detailed versions of a texture. MIP stands for multum in parvo, which is Latin for ‘multitude in a small space’ (Busby, Zak, & VanEenwyk, 2005); this is because a MIP map stores multiple texture images in one file. These maps replace the original texture as the player moves farther away from an object. Using MIP maps is a way of saving processing power, since objects further away
do not require as much detail. MIP maps also make an environment seem more realistic. As one moves closer to an object more detail in the surface becomes apparent, as in the real world (Linde, 2004).

Texturing Tools

Choosing which software to use when making textures depends on what type of textures needs to be created. Some programs are more suitable for certain tasks than others are; one program may be faster but another may produce results more in line with project requirements. As stated in the previous chapter, Photoshop was the only program used to create textures for this project.

Adobe Photoshop

Adobe's Photoshop (Adobe, 2009) needs almost no introduction. The name of the photo and image-editing program has become a common verb! Enter the word “Photoshop” into dictionary.com and a definition is found for what is a commercial product: “Photoshop (v). To edit an image using a computer program, 1992, originally in ref. To Photoshop, a bitmap graphics editor trademarked and published by Adobe, released in 1990” (“Photoshop,” 2009).

To say that Photoshop is industry standard would be an understatement, considering its name has become common vernacular. As the definition succinctly says, Photoshop is a bitmap image-editing program. Bitmaps, also known as raster graphics, are made of pixels, tiny dots of color. Raster graphics are an extreme example of pointillism, an art style where dots of color make up a larger image when viewed from far away. The nearly two-decades-old program has so many different ways to edit an image
that describing them all literally takes books to do. There are hundreds of brush options as well as the ability to create custom brushes. A layer system which not only allows one to stack images on top of each other but allows the user to apply 'layer styles' to those layers. Styles are details added to a single layer, such as embossing, or drop shadows. Photoshop is also fully integrated with graphics tablets, such as Wacom's Intuos, Bamboo and Cintiq lines. Since Photoshop is the graphics program of choice in the game industry, the program enjoys the support of many other programs. There are third party developed filters, such as NVIDIA’s normal map making filter. 3D modeling programs like Maya and 3DS Max automatically update saved textures while they are in Photoshop. Maya in particular also exports UV texture files as Photoshop files with specularity, bump, diffuse and other maps as separate, editable layers. When the Photoshop file is saved, all of the layers update in Maya.

 Modeling Tools

The process of modeling is not within the scope of this project; however, the choice of which 3D program to use is important because it impacts the speed with which one creates UV maps and UV textures. Some programs allow one to create UV maps more efficiently than others. For this project Lightwave 3D and 3DS Max were used for modeling and UV texturing. Silo v.2, Mudbox v.2 and ZBrush v.2 were used to test the creation of normal maps.

 **Lightwave 3D**

All the modeling for this project was done in Lightwave 3D (NewTek, 2009). Like all programs, it has good and bad points. On the good side, Lightwave 3D imports
and exports .obj files natively. Some programs do not do this, requiring the user to locate a plug-in in order to use .obj files. Another nice feature of Lightwave 3D is editing a model in Lightwave 3D does not result in having to completely recreate a UV map. On the bad side, the UV editing utility in Lightwave 3D is cumbersome. Basic operations such as the ability to merge points together are absent. Unfortunately Lightwave 3D is not popular within the modification and gaming community. Third party plug-ins are rarely developed for Lightwave 3D.

3DS Max

3DS Max is a 3D modeling program commonly used in game development (Autodesk, 2009a). Because 3DS Max is so popular, it is widely supported by third party developers and game modification communities. Individuals in these communities create tools for 3DS Max to aid the process of developing game content. 3DS Max is also has a powerful UV editing utility. The downside of using 3D Max is that the program has a fairly steep learning curve. I found that performing simple tasks such as splitting polygons in 3DS Max was frustrating. It was faster to model in Lightwave 3D and import models into 3DS Max for texturing. Another problem is that 3DS Max forces one to re-UV map an entire model if part of the geometry had been changed. This can be a problem if the model that needs to be remapped was complex, because one loses a significant amount of time while remapping.

Silo 2

Silo is a low priced modeling program that supports digital sculpting similar to ZBrush and Mudbox, while maintaining a minimalist interface and easy to learn controls (Nevercenter, n.d.). The program has a unique UV mapping utility that allows
the user to define on a model where to unwrap polygons for a map. It also supports exporting normal maps, which is why it was chosen for this project.

**ZBrush v.2**

ZBrush is a hybrid program where users can digitally sculpt and paint highly detailed models (Pixologic, 2008). Unlike modeling programs like 3DS Max or Lightwave 3D, ZBrush actually feels like one is sculpting with real clay. ZBrush uses imported images as well as 'paint brushes' to add texture to a model. The program also easily creates normal maps of models. The learning curve for ZBrush is steep though; its interface is unlike any other program on the market.

**Mudbox v.2**

Like ZBrush, Mudbox is a digital sculpting program that allows the user to create highly detailed models (Autodesk, 2009b). Unlike ZBrush, Mudbox is only a sculpting program; the user cannot add textures to the model, they can only edit geometry and create normal maps. If the user wishes to add texture to the mesh, they must export it from Mudbox into either ZBrush or a modeling program that can display textures. The interface in Mudbox is streamlined in comparison to ZBrush, making it easier to learn.

**Game Engines**

When making textures for games it is critical to periodically view those textures in a game engine as they are being made. Textures look significantly different once they are in-game. Engines add normal, specular and other maps to the texture, in addition to being affected by in-game lighting, video settings and player perspective. It is necessary to see these changes so that textures can be adjusted accordingly while they are being
produced. For this project several engines were reviewed as potential environments in which to display generated textures: Valve’s *Source* engine, Epic’s *Unreal 2* engine, Obsidian’s *Electron* engine and Bethesda Softwork’s construction set using the *Gambryo* engine.

**Bethesda Softwork’s TES: Oblivion – Gambryo**

I decided to use Bethesda’s *The Elder Scrolls: Oblivion (TES: Oblivion)* and its construction set *The Elder Scrolls: Construction Set 4 (TES: CS)* to create my digital environment (Bethesda Softworks, 2009). TES: Construction Set 4 was easy to use; models and textures only needed to be dropped into the correct folders and were imported into the game with the construction set. The ability to assign multiple textures to a model allowed for more detailed texturing results. The HDR lighting in *TES: Oblivion* enhanced the look of textures in-game. Lastly, the modification community for *TES: Oblivion* is large, active, informed, and happy to help solve problems.

The official Bethesda Softwork’s release *TES: Oblivion* is part of Bethesda Softwork’s *The Elder Scrolls (TES)* mythos. Several games have been produced over the years shaping the pseudo medieval fantasy world of Tamriel. Old stand-by monsters and races are mixed with others unique to *TES* world, creating a familiar but engaging environment.

**The Construction Set**

TES: Construction Set 4 is the tool used to modify *TES: Oblivion*. In TES: Construction Set 4 one can import models, textures, build environments, and then populate them with people. The game is powered by the Gambryo game engine, licensed
to Bethesda Softwork’s by Emergent Game Technologies (Emergent Game Technologies, 2009). Integrated into the engine is the Havok physics simulator, while foliage is generated by a third party program called Speed Tree. To test textures and create the example digital environment, it was necessary to learn how to use TES: Construction Set 4.

Summary

Thanks to consumer and development demands, graphics technology has rapidly progressed forward in the past few decades; however this rapid march forward has lead to several different, sometimes contradictory, methods of texture creation. Which can make learning how to texture challenging.

Chapter III focuses on the creation of texture samples and testing of questions about the texturing process based on research done in Chapter II. The methods of texturing previously discussed in this chapter are used to develop texturing samples. These samples were viewed within a digital environment and presented to the public in the form of a survey. Feedback from the survey was compared to my observations about the samples. The results from these comparisons were decisions about how textures were made; which were then implemented in the form of a custom digital environment.
CHAPTER III

ANALYSIS OF TEXTURING TESTS

Introduction

Chapter III discusses the creation of texturing samples used to determine effective texturing techniques for video games. The first section of this chapter will cover how samples were developed and creation of the online survey instrument. The second section of the chapter will cover the analysis of survey test results and the impact they would have upon the creation of the digital environment.

Development of Texturing Tests

The texturing samples were created to be a way of gauging what texturing procedures produced the best visual results. However, before samples could be created constraints had to be placed upon how samples were generated. All textures were generated in Photoshop without the use of photographs. This was a stylistic decision; I did not want textures to look photorealistic in this study. Textures were created in three sizes: 512x512 pixels, 1024x1024 pixels and 2048x2048 pixels. 512x512 pixels was the size most recommended by researched sources, with 1024x1024 pixels a ‘high end’ size and 2048x2048 pixels considered excessive (Franson, 2004). The texture sample generated was a wood grain. Wood is an easily recognizable texture, so it would be theoretically easy for survey participants to spot if something about it was off. The
programs used to create normal map samples were ZBrush, Mudbox, Silo and Photoshop. All textures and normal maps were tested in a blank world made in the TES: Construction Set 4 and viewed in TES: Oblivion.

With these constants established, I created four methods of texture generation designed to answer questions I had about the texturing process (see Chapter I: Purpose of Study). The samples created from these methods of texture generation would assist me in deciding which of the methods produced the highest quality texture results.

**Painting at Size vs. Rescaling**

Painting large textures and scaling them down to intended size was a throwback from the first generation of 3D video games when texture sizes had to be very small (see Chapter II). Many of the researched sources recommend this practice to novices (*The Darkside of Game Texturing* and *3D Game Textures*). The reasoning for this practice was that painting on a large canvas was much easier than painting on a small one. Some people even insisted that they could pack more detail in a small space if they scaled down from a large image (Franson, 2004). Other artists disagreed with this practice and warned that any resizing will degrade image quality (Mathis, n.d.). The first set of samples was created to determine if textures at intended size were of significantly higher visual quality in comparison to those that were scaled down to intended size. Nine texture generation methods were used to create the samples textures:

- Painting at intended size.
- Scaling the image to intended size.
- Scaling the image to intended size and retouching.
- Cropping the image to intended size.
• Cropping the image to intended size and retouching.
• Applying a smart object layer effect then scaling the image to intended size.
• Applying a smart object layer effect then scaling the image to intended size and retouching.

These samples were created in three texture sizes, 512x512 pixels, 1024x1024 pixels and 2048x2048 pixels. Each of the resizing operations recalculated the position of pixels in a texture when they were preformed and thusly, each produced slightly different end results. The results were placed in a test world and applied to large models called billboards and arranged into lines by size. All textures of size 512x512 pixels occupied the first line, textures sized 1024x1024 pixels occupied the second line and 2048x2048 pixels the third line (see Figure 6).

*Figure 6. Billboards.*
All texture samples were generated from a process I developed to make wood grain (see Appendix D). Twenty-one texture samples were generated for comparison.

**Tiling Textures vs. UV Textures**

There are two predominant styles of texturing that have carried over through the years, tiling textures and UV mapping textures. Deciding between when to use a UV texture or a tiled texture depended on how important the object was and where it was placed in game. The tradeoffs between using tiling textures or UV textures were visual quality and time. UV textures look less generic, but take longer to make. Tiling textures are faster to generate and fill large amounts of space, but can make an environment look less visually engaging. The two texturing styles were applied to differently shaped models, and the visual results of the textures were compared. The samples created were:

- Tiled texture barn at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
- UV texture barn at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
- Tiled texture sphere at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
- UV texture sphere at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
- Tiled texture lion statue at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
- UV texture lion statue at sizes 512x512 pixels, 1024x1024 pixels, and 2048x2048 pixels.
The barn represented box shapes, the sphere represented round shapes and the lion statue represented irregular organic shapes. All models were lined up in rows according to texture size as well as texture type. For the barn it was decided to apply the texture only to the sides of the barn model but not the roof. The intended-size textures from the resizing samples were used as tiling textures because the process was the same for creating them. The UV textures were made entirely by scratch because they had to be fit into a UV map. In the test environment, barn models were arranged in ascending order from 512x512 pixels to 2048x2048 pixels starting with tiling samples followed by the UV samples. The barns were row one (Figure 7). Behind the barns were the sphere shapes (Figure 8). Behind the spheres were the organic lion statues (Figure 9). Lining up the models allowed for faster comparisons between samples.

Figure 7. Barns.
Figure 8. Spheres.

Figure 9. Lions.
Generating Normal Map Samples

Normal map samples were created using four programs to find which program produced the best results with the least amount of experience. The four programs I used to make samples with were:

- ZBrush v.2
- Mudbox v.2
- Silo v.2
- Photoshop CS2 using the NVIDIA normal map filter

It was decided to focus on making simple marks for the normal maps: a hard-edged square and a soft-edged circle (as seen in Figure 10). The reasoning was that simple marks would be just as clear an indicator of normal map quality as complex marks. The normal maps were applied to cubes with a generic texture in order to be easily compared.

Figure 10. Normal map cubes.
The Survey Instrument

A survey was created to validate findings from the test results. Comparing personal opinion to survey results would lend more weight to the findings. The survey results would impact the creation of textures for the example digital environment, specifically relating to the size and type of textures used. Surveying other people would indicate whether viewers could see differences in texture quality or not. This information is useful because if viewers could not see differences between large detailed textures and small not-as-detailed textures, those small textures could be used without the viewer noticing. Knowing this is important, because if small textures take considerably less time to make in comparison to large textures, an environment can be quickly made with smaller textures without apparent loss of visual quality.

An online survey was used to present tests instead of running viewers though the game in person. An online survey was faster to distribute and capable of reaching far more participants. The websites where links to the survey were posted were: Hanover College Online Listing of Psychology Surveys (Krantz, n.d.), the Penny-Arcade Enforcer Forums (a personal art site), and a post on the Game Developers Conference - Conference Associate (GDC – CA) mailing list. Penny-Arcade is a popular web comic which focuses on video games and the culture that surround them. The Enforcer Forums are where people who work at the Penny-Arcade convention every summer keep in touch. The GDC is the Game Developers Conference held in San Francisco every spring, a huge convention for people who develop video games. The CA mailing list is a resource for those who work at the convention, the Conference Associates, to keep in touch. These places would reach a potentially large audience from a variety of
backgrounds. The Penny-Arcade forums and the GDC - CA mailing list provided people who were familiar with video games while the Hanover College and art site provided those who may not be as familiar with games.

The survey instrument was developed in Adobe Flash (previously Macromedia Flash). The survey began with an introduction, followed by an informed consent page, then a brief explanation about the aims of the survey and what participants would be asked to do. Upon agreeing to the conditions, participants were asked to provide non-traceable personal information: Age, gender, if they played games, if they were a student, occupation, if they had ever worked in the game industry and if they had an art background. After answering the opening questions, participants were taken to a series of screens that featured a large image of a texture taken from the test environment and four questions asking them to rate the texture. Questions were rated on a scale of one to seven, one being the lowest score, seven the highest. Participants were asked to examine a texture sample then rate it on the following criteria: was the texture blurry? Did the texture display noticeable banding? Was the texture pixilated? How attractive was the texture? For normal maps, participants were asked to rate based on: Was the normal map clearly visible? Did the normal map look like it was painted on or 3D? Did the normal map make the object appear to be more detailed and complex? Was the normal map attractive? Forty-one screens were created, with four questions per screen, and a short two screen break near the end explaining what a normal map was so participants would know what to look for when rating samples (see Appendix C for survey samples). After two days, the number of screens was cut from forty-one to thirty-two. Numerous participants emailed in protest of the survey length, warning that most
people would not complete the survey in its entirety due to the amount of time it took. In response to this feedback, results from the resized and touched up tests were removed. These tests were removed because they were deemed redundant. In order to use all the surveys, the results from the touched up texture tests were thrown out from the survey feedback turned in during those two days. Once participants were finished with the survey, the results were anonymously emailed from the website to an email account and the values recorded in a spreadsheet. The survey ran for a month, with over forty-nine responses gathered over a wide range of ages and occupations.

**Survey Results Analysis**

Each set of samples was analyzed to determine if they had produced what could be considered high quality results. Participant reactions to the survey results were then reported and compared to personal observation. Survey participants were asked to rate textures based on a number of criteria. These criteria were:

- Was the image was pixilated?
- Was the image was blurry?
- Did the image displayed banding in the texture?
- Was the participant attracted to the texture?

Normal maps were rated based on how well they simulated the illusion of 3D. The criteria participants were asked to rate normal maps with were:

- Did the normal map seem to add complex detail to a model?
- Did the normal map appear flat on the model?
Did the normal map appear painted onto a model?

Was the participant attracted to a normal map?

Intended Size Texture vs. Scaled Texture Tests

The first set of textures samples sought to discover if textures would be of better visual quality if they were painted at intended size or painted large and scaled down to intended size. It was expected that large textures would not lose image quality when scaled down, at any size.

Personal Observation

Many of the literary sources researched asserted that resized textures were of the same visual quality as those painted at intended size. However, when comparing the resized texture samples to intended size samples, all the scaled samples were slightly blurry no matter which resampling had been done. When looking at the rescaled with touch up (which was removed from the survey after two days) the images were blurry even after a sharpen filter had been applied to the texture.

Personal opinion about the intended size vs. resizing tests was that larger textures were crisper and looked smoother in comparison to the smaller textures. The visual differences between incremental sizes were negligible. The 512x512 pixel and 1024x1024 pixel textures were practically interchangeable when looking at intended size tests. The same observations were made of the 1024x1024 pixel and 2048x2048 pixel sized textures. Only when comparing the 512x512 pixel to the 2048x2048 pixel tests, did one see a significant difference in visual quality.
Survey Participants Feedback

The following graph (Figure 11) shows averages of survey feedback for the intended size and resizing tests. The horizontal rows tell us the file size and type of test being performed while the vertical columns tell us the survey questions asked and the responses to them.

Figure 11. Graph of resizing survey results.
The overall participant rating of the resizing tests was in favor of the 512x512 pixel intended size test. In fact, as texture size went up, scores dove down; as seen in the response average of intended size vs. resizing tests graph. All the resize tests scored lower in blurriness when compared to intended size textures of the same size.

Summary

Working at intended size meant additional editing was unnecessary. Textures were tighter, crisper and time was saved because there was no need for touch up once the texture was done. While larger textures looked better to me, participants preferred small textures over large ones. It was concluded that using small textures was desirable when trying to save on file size but using textures sized 1024x1024 pixels was more visually appealing. Discussion of possible reasons why participants preferred the smaller textures are discussed at the end of the chapter.

Tiling Texture Tests

Round two of texture samples examined the size of tiling textures across three types of models. It was expected that the larger textures would always have significantly more appealing visual quality in comparison to smaller textures across all model types.

Personal Observation

In order to save time, the intended size samples from the previous sample set were applied to differently shaped models. This was done to observe if tiling texture results across three differently shaped models would produce acceptable results, as well as if texture size impacted visual quality.
Unlike the previous samples, visual quality between textures was more pronounced across each model type. The box shapes are all within acceptable limits in texture quality, but each jump in texture size was a noticeable increase in quality. However, this was if one is standing very close to a model. When viewed from a distance, the 512x512 pixel and 1024x1024 pixel textures had no significant difference, nor did the 1024x1024 pixel texture when compared to the 2048x2048 pixel texture.

All texture sizes tiled across the sphere awkwardly, cutting off in strange places, or distorting the texture. The larger textures were of higher visual quality, but were brought down by the unnatural way the texture tiled across the spheres surface. The lion looked similar in each instance; in fact, it looked better than any other test shape. The larger texture did seem to have more definition, but the smaller texture just seemed to be made of a rougher wood. However tiling a texture across an organic shape was awkward, with the texture cutting off at strange, angles. It seemed tiling textures needed to be applied to blocky shapes, not round or organic ones.

Survey Participants Feedback

Figure 12 shows participant responses to the tiled texture tests. The horizontal values represent the size of texture being tested, as well as the type of model the texture was applied to. The vertical values represent the questions asked of participants and responses to the test.

The audience feedback about the tiling tests is interesting, but needed to be taken with a hefty grain of salt. The first test, the 512x512 pixel texture applied to the barn, received above average scores, but for each larger size of texture applied to the barn, scores dropped. The scores for the sphere were inconstant, with the 512x512 pixel
<table>
<thead>
<tr>
<th>Texture Size</th>
<th>512 tiled</th>
<th>512 tiled</th>
<th>512 tiled</th>
<th>1024 tiled</th>
<th>1024 tiled</th>
<th>1024 tiled</th>
<th>2048 tiled</th>
<th>2048 tiled</th>
<th>2048 tiled</th>
</tr>
</thead>
<tbody>
<tr>
<td>barn</td>
<td>4.7</td>
<td>5</td>
<td>3.7</td>
<td>5.4</td>
<td>2.1</td>
<td>5.6</td>
<td>4.7</td>
<td>3.8</td>
<td>5.1</td>
</tr>
<tr>
<td>sphere</td>
<td>5</td>
<td>4.5</td>
<td>3.9</td>
<td>5.5</td>
<td>4.5</td>
<td>4.5</td>
<td>5</td>
<td>4.3</td>
<td>4.4</td>
</tr>
<tr>
<td>lion</td>
<td>6</td>
<td>4.5</td>
<td>4.5</td>
<td>5.5</td>
<td>5.5</td>
<td>5.1</td>
<td>4.9</td>
<td>4.8</td>
<td>4.6</td>
</tr>
<tr>
<td>Attractive</td>
<td>4.6</td>
<td>4.5</td>
<td>2.9</td>
<td>4.8</td>
<td>2.4</td>
<td>4.1</td>
<td>3.5</td>
<td>2.6</td>
<td>3.8</td>
</tr>
</tbody>
</table>

**Figure 12.** Graph of tiling test survey results.

sphere texture rated highest, the 1024x1024 pixel texture scored the worst, and the 2048x2048 pixel texture rated slightly higher than the 1024x1024 pixel texture. Strangely enough the 1024x1024 pixel texture on the lion statue scored higher than both the 512x512 pixel and the 2048x2048 pixel textures.

**Summary**

There were some fundamental disagreements on texture quality between what I observed and survey participants. In the barn test, participants indicated that they preferred the smaller texture instead of the larger. To one whom has been trained in
creating textures, this was backwards. It was expected that the smaller sized texture would have scored lower than the two larger ones. The only scores agreed upon were for the organic shape, with the smaller texture scoring lower than both the larger textures.

For these samples, compromise was the best course of action. It was personally felt that 2048x2048 pixel textures provided the best texture quality over all tests, but participants disagreed preferring either 512x512 pixel textures or 1024x1024 pixel textures. Using the 1024x1024 pixel textures was an acceptable compromise considering file size and visual preferences. Tiled textures were still a necessary part of the production process and an effective way of creating crisp clean textures. The shape of the object needed to be considered, tiling was visually poor across a sphere or an organic shape, no matter what the texture size was.

UV Texturing Samples

The third set of samples sought to discover if the size of a UV texture would affect its visual quality when compared across different models. Again it was expected that large textures would be significantly higher in visual quality when compared to small textures.

Personal Observation

UV textures showed drastic differences in visual quality between tested sample sizes. The 512x512 pixel sized UV textures were extremely blurry, so much that it was impossible to determine what the texture was supposed to represent. The 1024x1024 pixel texture was not much better, though the texture was slightly easier to recognize. The 2048x2048 pixel texture was the only one that was distinguishable as
wood. The size and shape of the object greatly affected the look of the texture. On the large models, the barn and sphere, textures were stretched and blurry, drastically reducing the visual impact of the model. On the smaller model, the lion statue, the 512x512 pixel texture was of noticeably lower quality, the 1024x1024 pixel and 2048x2048 pixel textures were acceptable.

Survey Participants Feedback

Figure 13 shows participant responses to the UV texturing tests. The horizontal information is texture size and type of model the texture was applied to; the vertical represents the questions asked of participants and the responses given.

Figure 13. Graph of survey responses to UV tests.
The 512x512 textures scored the lowest in all categories, with the 1024x1024 pixel textures scoring marginally higher than the 512x512 pixel textures and the 2048x2048 pixel textures scoring slightly higher than the 1024x1024 pixel textures. The 1024x1024 pixel lion texture scored the highest of all the UV tests.

**Summary**

In general, the survey participant feedback was in line with my observations of the texturing samples. The small sized textures looked worst on large models, but acceptable on small ones. I did not agree that the highest quality texture of the sample group was the 1024x1024 pixel texture on the lion statue. However, I decided that a 1024x1024 pixel texture was an acceptable compromise between file size and potential time spent to make it.

Applying UV textures to large objects did not yield visually appealing results, which was backed up by survey participant feedback. Small objects could have small UV textures, because small textures did not look stretched and blurry on a small object. If a large model needed to be UV textured the best-case scenario would be to break up the model into pieces and UV texture each piece, instead of having one large model on a single UV map. Texturing this way gives each piece of the model as much space as possible in order to be as detailed as possible. However, texturing objects this way increases the amount of textures needed to be produced, effectively doubling or even tripling an artists’ work load.
Tiling Textures vs. UV Textures

Figure 14 shows averages of responses from both the tiling and UV texturing tests. Feedback was averaged for each test to make the graph easier to read. Averaging test responses was done to get an overall feel for participant reactions to each test.

![Figure 14. Averages of responses of UV and tiling tests.](image)

Overall tiled textures were rated higher, especially when comparing like sized textures to each other. For example when comparing a 512x512 pixel tiled texture vs. a 512x512 pixel UV texture, the tiled texture was always better received. However, this only held true across large models, it did not hold true for the smaller lion statue. The UV texture for the 512x512 size was not as high of quality as the 2048x2048 pixel texture but it was still easily recognizable. Discovery that polygon size would affect textures was a
significant and unexpected find which impacted how models were made for the digital environment.

Normal Map Samples

The fourth set of samples sought to find which program would produce better normal maps with the least amount of user experience. Tests were performed in four programs: ZBrush v.2, Mudbox v.2, Silo v.2, and Photoshop CS2. It was expected that hybrid programs such as ZBrush and Mudbox would produce the best results, but take in the longest time to get useable results from. Silo and Photoshop were expected to produce lower quality results.

Personal Observations

Learning ZBrush enough to simply move around within the interface took several days. The program is so different from any other 3D editing software that one must learn the terminology specific to ZBrush before creating objects in it. Mudbox was easier to learn; in fact, normal maps were being produced within an hour of using the program. Silo was even easier, with normal being exported from the program within minutes. Creating normal maps in Photoshop was similarly easy, but creating normal maps which would show up in the game engine took a few days of testing and researching.

ZBrush produced the highest quality normal map results. The maps were crisp, clean, and easy to see once in-game. Normal maps were created using a technique learned from an online tutorial which made painting a normal map simple (Ashlock, n.d.). This technique was adapted to make normal maps in Mudbox as well, which again,
made it easy to create the normal map. Normal maps produced in Mudbox were acceptable, but not as effective as ZBrush. Edges meant to be sharp were softened while intentionally soft edges were so subtle they were nonexistent. The program also made square indentations round. The normal maps produced in Silo barely made any visual changes to the model in the game engine. The indentations were so faint they were barely visible. The program also distorted the normal map on export, turning it into an image faintly reminiscent of what was originally created. Producing normal maps in Photoshop was, in comparison to the other programs, easy and quick. Creating a normal map in Photoshop literally took seconds, by running the normal map filter; yet that did not mean the results were useable. One needed to have a better understanding of what a normal map was in order to generate usable results. Even then, normal maps were not as sharp as those created in either ZBrush or Mudbox. But considering the speed with which the Photoshop normal maps could be created and the amount of power one had over them in the program, they could not be dismissed out of hand.

Survey Participant Feedback

Figure 15 is the response averages of participants to the normal map tests. The horizontal values are the programs tested, while the vertical values are the questions asked of participants and the given responses.

ZBrush scored the highest, Mudbox second, Photoshop third and Silo last. Silo actually scored higher than was predicted.
Figure 15. Graph of survey responses to normal map tests.

Summary

The audience agreed with my observations that ZBrush produced the highest quality normal maps. Mudbox was rated second, with Photoshop and Silo rated last. I felt that Photoshop normal maps were significantly better than those of Silo.

ZBrush was the most capable when it came to making normal maps and exporting, but had a steep learning curve. Photoshop was the easiest to make normal maps in and was much more suited to making normal maps for tiling textures. Yet making normal maps in Photoshop required more knowledge about what a normal map was in order to create effective ones. Just running the normal map filter resulted in
useless normal maps. Mudbox normal maps were effective, but having the option in ZBrush to make a texture while making the normal map was invaluable. Silo is primarily a modeling program; while it could be used to export normal maps, it was better to export its models into either Mudbox or ZBrush to make normal maps.

Limitations of Survey

The results of this survey needed to be scrutinized carefully. None of it could be taken at face value due to several limitations. The first issue was the length of the survey. During the first two days the survey was available participants whom had completed it emailed me to say that it was much too long and would drive other participants away. These warnings were backed up by the fact that several of the returned surveys, after the first twenty-one questions, were all scored ‘one’ for every question. ‘One’ was the default answer for the questions, which told me that people were simply clicking the ‘next’ button to finish the survey quickly. Participants also said that questions were confusing and that they could not tell the difference between presented test images. This confusion could have been caused by lack of a reference image to compare textures to during the test. However, not giving participants an image to compare test results to was an intentional decision; I wanted to see what participants reactions to textures would be if they did not know what a “good” texture looked like. Unfortunately, this idea backfired, only serving to bore and confuse participants, skewing survey feedback in the process.

During the course of the survey, it had been hoped to compare responses of those who worked in the game industry to those who did not. I had wanted to see if those
who worked in the industry would be more aware of texture sizes as opposed to those whom were less familiar with the game making process. Unfortunately, I received only six responses from people whom indicated they were working or had worked in the game industry. Looking at those six responses revealed that they were no better or worse than those who claimed to not work in the game industry. However, in order to draw more useful information from the survey, many more responses would be needed from those in the industry. In a similar thread, it was hoped to compare responses of gamers to non-gamers, but the overwhelming majority of people who responded to the survey were gamers of all ages. While this came as a bit of a shock, it was necessary to remember that ‘gaming’ is no longer a niche past time, despite popular myth. Gaming includes all types of people now, not just those who live, breathe, and eat games. This was interesting because the search for survey participants was split between gamer heavy sources, the GDC-CA mailing list and Penny-Arcade enforcer forums and what were considered gamer light sources, Hanover psychology research website and an art website. The possibility exists that those attracted to the survey from the ‘gamer light’ sources may have been gamers who frequented those places. Perhaps those with no interest in games passed the survey over.

There is much about this survey which would be changed in order to collect data more effectively. It would be streamlined, with fewer screens and questions for participants to answer. The resizing tests would be separated from the model tests in order to reduce the amount of time participants would need to spend on the survey. The drawback to this would be getting a disproportionate number of responses on one survey over the other. A reference image would also be added for participants to compare to the
test image. One question would be added to the general information gathering, asking those who gamed how much time a week they spent playing games. This question would be added as a way of separating the heavy and light game players to better compare responses.

Summary

According to the results of this study, 1024x1024 pixel tiling textures were the best looking textures overall, and a good compromise in file size. Textures needed to be painted at their intended size, because resizing lowered image quality. UV textures should be used on small to medium objects and large models should be broken down into multiple UV maps. In order for a UV texture to have an acceptable visual quality it should be as large as possible and painted at size to avoid lowering image quality. If possible, normal maps should be created in ZBrush, as the program produces the highest quality normal maps of the four programs tested. However, Photoshop should be used if speed is an issue or if the normal map is intended for a tiling texture.

Impact on Digital Environment

Textures for the example digital environment were created based on survey results and my observations. Texture sizes were kept between 256x256 pixels and 512x512 pixels for small objects, and 512x512 pixels and 1024x1024 pixels for everything else. Most of the prop objects were UV textured, which caused them to take longer to make, but the results were better than simply tiling them and hoping for the best. Tiled textures were between 512x512 pixels and 1024x1024 pixels, with the 1024x1024 pixel textures applied to very large objects; the walls and ceilings of the
house and barn for example. Models made for the digital environment were altered to reflect discoveries made during the testing phase. It was found that textures looked better on models with higher polygon counts; therefore, extra geometry was added to some models. Both the house and barn models were split in half to add more polygons and allow textures to tile cleanly. Many models, not just the large ones, had several textures applied to increase their overall level of detail.

Chapter IV describes the example digital environment created to demonstrate texturing techniques developed for this project; from preproduction of the project to the creation of assets.
CHAPTER IV

METHODOLOGY

The Digital Environment

The scope of the environment had to be small enough that it was possible for a single person to complete in a reasonable amount of time. That meant that the story and the environment had to be kept very simple. It was decided instead to focus on creating everything the game could not provide the project. TES: Oblivion is a medieval fantasy game; therefore, it lacks modern content, so Twentieth century content would have to be created.

Preproduction

Preproduction is preparatory work done before a project goes into full production. In the preproduction phase, stories are written, puzzles developed, concept art painted and storyboards drawn. Other media were sought out for story inspiration; books, games and TV shows were my primary sources with movies and comic books my secondary sources. The subject of the paranormal or the fantastic in everyday life was the theme of the project, so media focusing on the subject was sought out. Fortunately, the subject was popular, making research materials easy to find.

Ghost Hunters, a show on the Sci-Fi channel, is interesting because it was about ordinary people looking for the fantastic in the real world (Scifi Channel, n.d.). Many of the environments investigated were old abandoned buildings, places too
dangerous to visit alone. Using this concept would provide ample opportunities to create a variety of textures. Choosing an abandoned location was also homage to titles such as *The 7th Guest* (Mobygames, 2009a) and *Myst* (Cyan Worlds, n.d.), games that have shaped my perceptions of what a game experience should be.

When looking at other video games, one of the most influential series available in terms of atmosphere and storytelling is the *Silent Hill* series (Mobygames, 2009b). These games are a Japanese twist on American horror movies. Unlike American slasher horror, fear in these games is created though oppressive atmosphere and tense storytelling. The games seek to make the player anxious and always wondering what those noises coming out of the darkness were. Each game in the *Silent Hill* series consists of two sides, a “Real World” and an “Other World.” The Real World is a lonely abandoned town perpetually covered in fog. The Other World is dark, gritty, and disturbing. Walls are slick with unidentifiable red-brown substances; the ground often nothing more than metal grating hanging above gaping pits. This environmental duality served as both a game play and storytelling mechanic.

I also looked to Neil Gaiman, a fantasy and graphic novel writer, for inspiration because much of his work deals with the theme of duality and fantastic in the real world. In particular, the story of *Coraline* (Gaiman, 2002) both for it’s setting and how it treats the theme of duality.

Drawing from these sources, I developed a small story. The setting was an old abandoned farmhouse, with a mirror world though a door in the basement. The abandoned home was worn and run down, the mirror world was new looking but desaturated nearly to the point of being in black and white. In order to effectively concept
the location, I needed a reason why the nonexistent player was there. Taking a cue from *Ghost Hunters*, the player became an urban explorer investigating an abandoned house with a grim past. While exploring the house, this person would discover a large foreboding door. Past the door would be a long dark passage at the end of which was yet another door and the mirror world. In order to escape the explorer needed to search the mirror world house to find a key to unlock the now mysteriously locked basement door. From this setting storyboards were created.

**Storyboards**

Storyboards are commonly used in the film and game industry to plan out a story before either filming or production takes place. Storyboards save precious resources by allowing experimentation with cameras and pacing to take place before committing them to film. Storyboards also serve as a form of communication between members of a production team. Since this was a one-person production team, the use of storyboards took on the role of concept art. The storyboards depicted placement of objects, lighting, and experimentation with props before creating them in digital form. Fifty-one pages of pencil-sketched storyboards were produced following the explorer though the abandoned house (see Appendix A).

From the storyboards, model and texture lists were compiled (see Appendix B). The model list documented every model that would be created in order to flesh out the environment: furniture, large objects like the house, the interior walls and floors, doors and decals. The initial model list was over sixty items long, with between one and six textures associated with each model. The texture list compiled all the textures needed
and noted where textures could be reused. Once model and textures lists were completed, full production began.

Learning the TES: Construction Set

The process of creating a digital environment was in no way linear. There was a great deal of starting, stopping, and backtracking. The greatest task was learning to create an environment within the TES: Construction Set 4. The construction set for TES: Oblivion had several utilities built into it to facilitate creating custom digital environments or altering the official game. To make worlds separate from the official game one needed to create new world spaces (the exterior landscape), new interiors, manipulate the land of the world and then populate it with objects. Accomplishing these feats first required learning how to maneuver within the TES: Construction Set. Seeking out tutorials online was the most expedient course to learn how to use the TES: Construction Set, due to a lack of formal information. These tutorials were written by other people whom were also learning how to use the TES: Construction Set 4. The most basic of the tutorials taught how add to the world of TES: Oblivion. For example, the first tutorial I read covered putting in a house and giving it a simple interior (“The Elder Scrolls Construction Set Wiki,” n.d.). Other tutorials taught how to make dungeons, setting up lights in game, changing the landscape of the game, dealing with creature AI and adding models and textures into the engine (UESPWiki, 2007).

This Old House

In order to create environments with custom content, models had to be prepped for importing into the game engine. Preparatory work included importing models
into 3DS Max, creating collision meshes, texturing, normal mapping all while testing models at each step. Using the house model as an example, we will go through the process of creating custom content, starting with exporting the models from Lightwave 3D and ending with the finished product as it would be seen in the game engine.

Working from the storyboards and a floor plan, the exterior and the interior house models were built in Lightwave 3D. The first model built was the exterior of the house, looking to references of New England style farmhouses for architectural details such as the front porch, roof, and fireplace. Once the exterior house model was built, the interior house model was created within the finished exterior in order to maintain continuity of size between exterior and interior spaces.

Four models comprised the house; the exterior house, a basement, an attic, and the first and second stories. As the models were finished, they were exported from Lightwave 3D and imported into 3DS Max. Once in 3DS Max they were resized and exported as .nif files to be tested for scale in the game engine. The .nif file is the only file format the game engine recognizes; therefore, all models had to be converted to .nif files before they could be loaded into the game engine. The models were added to the game with the TES: Construction Set 4 dropped into a test world space and saved. In TES: Oblivion models were inspected from player perspective within the game. If the models were too big or small they were resized in 3DS Max and re-exported to be tested again. Once the models were sized correctly, collision models were added in 3DS Max to the main models.

Collision models were copied versions of the main models. For the exterior house model three separate collision meshes were used, including stone, wood and stairs,
which triggered the player character to walk up the stair collision mesh. The first and second floor model had two collision meshes, one for the floors and walls, the other for the stairs. The basement model also had two collision meshes; one for the walls and floor, the other for stairs. The attic was the only model that did not have multiple collision meshes. Each time a new collision mesh was added to a model, it was immediately tested in-game to make sure player characters did not fall though the floor or walk through walls.

Once the house and collision models were created, sized correctly and working in-game, texturing could begin. Only final versions of models were textured because 3DS Max forced the user to recreate UV maps if models were altered. If a UV map needed to be recreated, one lost a lot of time, especially if the UV map was complicated. The outside of the house needed to be split into several different textures: first floor walls, second floor walls, trim between the floors, trim between the second floor and the attic, wood under the roof, trim under the roof, the porch, the porch supports, the porch floor, the roof, the fireplace and the steps leading up to the house. Some portions of the house could reuse textures, but others such as the concrete steps, needed to be unique. Sections of the model were detached and renamed in order to receive unique texture properties. Depending on the piece being textured different approaches were taken. Some objects were treated as tiled textures; for these models, the UV map was flattened, straightened and polygons were moved so the texture did not display upside down. For example, the interior and exterior walls of both the barn and house were tiled in this manner. Other models, ones that needed unique texture maps, were UV mapped. A maximum of six textures could be applied to any one model.
Applying more would crash the TES: Construction Set when trying to load the model. Three wood textures were made for trim, siding, and the porch. Unique textures were created for the roof and steps, and a tiling stone was made for the first floor walls.

Texturing was a great deal of trial, error, luck, and patience; even after testing texture creation methods (see Chapter III). Many textures were recreated several times until a useable one was finally settled upon. For objects that were going to be in both the Real World and the Mirror World, two textures were made. One texture represented the Real World while the other texture represented the Mirror World. The two textures were made simultaneously in the same Photoshop file, separated by folders. The Mirror World texture was created first; this texture was supposed to look unblemished, except for the fact that the texture would be nearly black and white. Color was removed from the image by using a desaturation layer effect on top of all of the other layers, a nondestructive way of altering the image (see Appendices E and F for non-destructive texturing techniques). Once the Mirror World texture was finished, a new group of layers was added in Photoshop. These new layers were the grime layers. Grime layers are painted overlays simulating scratches, dirt, stains, and other discolored nastiness. Once the texture was finished, it was saved as a .dds file. The .dds format compresses images as small as possible whilst retaining as much detail as possible. The NVIDIA .dds Photoshop plug-in provided several different types of .dds file formats to save an image, along with an option to save MIP maps as well. Two .dds formats used throughout this project, the DTX1 RGB and the DTX5 ARGB formats. The difference between these two file formats was whether an alpha channel was present in the file. In TES: Oblivion the presence of an alpha channel in a texture created transparency in the game. DTX1 RGB
file compression removed alpha channels from a texture, while DTX5 ARGB added an alpha channel. For example if one made a glass bowl, its texture would be saved as a DTX5 ARGB file leaving the alpha channel intact so that the bowl was transparent in-game. All other textures, such as wood or stone were saved as DTX1RGB files. Normal maps could have alpha channels as well. Normal map alpha channels were how specularity maps were included in TES: Oblivion. All textures applied to the house were saved as DTX1 RGB files. The stone wall normal map used a specularity map in order to prevent the wall from appearing like plastic.

Once the texture was completed, a normal map would be made from the texture. To make normal maps two procedures were used; one in Photoshop the other using a combination of ZBrush and Photoshop. The Photoshop method was used to create the majority of the normal maps, as was determined by the normal map samples created and discussed in Chapter III. ZBrush was used for the normal maps that needed to be more pronounced.

Normal maps created in Photoshop began with a copy of an existing texture converted to grayscale. When a texture was converted to grayscale, the dark places and the light places would not necessarily coincide with areas of height and depth on the model. For example, a dark stain on a texture would also be a dark mark on the grayscale image. Left unedited this dark stain would become an indentation on the normal map. This indentation may be in an undesirable place, may be distracting, or may not be representative of the surface it was applied to. Thus, the grayscale image was edited to remove these potentially distracting areas. Contrast in the grayscale image was increased drastically, dark areas were made very dark and light areas made very light. It was
discovered that more contrasted grayscale images produced stronger normal maps. Once
the grayscale image was adjusted, the NVIDIA normal map filter was run. The default
settings were used to enable an accurate normal map. The filter converted the grayscale
image into a pastel blue, green, and purple image. In order for the normal map to work
properly in *TES: Oblivion*, the red and green channels were swapped in Photoshop. Not
switching the channels caused the normal map to display inverted in game. Once the red
and green channels were switched, the overall image was inspected; if the image did not
to have enough contrast individual channels could be edited further. The only
adjustments made to these channels were repeatable functions that could be applied
equally to each channel, such as brightness/contrast or a curves function. Unequal editing
of channels created inconsistencies in the normal map, and caused the map to not display
correctly within the game engine. Major adjustments beyond increasing channel contrast
needed be made during the grayscale stage, when one was working with a single channel.
A more in-depth explanation of this process can be found in Appendix G, a tutorial for
creating normal maps in Photoshop. When finished the normal map was saved as either a
DTX1 RGB or DTX5 ARGB .dds file with a '_n' suffix at the end of the filename. The
suffix told *TES: Oblivion* the file was a normal map and should be displayed with the
texture of the same name. For example, the texture file 'house.dds' would have a normal
map named 'house_n.dds'. Both the texture and the normal map were placed in the
appropriate texture folder in the Bethesda *TES: Oblivion* files and were associated
automatically with the model when it was imported into the construction set. The model
with associated textures could then be viewed in-game and adjustments to the texture
made in Photoshop.
ZBrush is typically used for creating unique models that need to look complex but have low polygon counts. However, ZBrush can also be used to create tiling textures, it just requires planning ahead. The ZBrush method works backwards in comparison to the Photoshop method, starting with creating the normal map then making a texture based on the normal (Ashlock). This method was used to create the stone wall texture of the house and stone wall surrounding the environment. To create the stone wall normal and texture, a two dimensional object with no depth called a plane was generated in ZBrush. To this plane was assigned a ZBrush material called RGB that converted the plane into a normal map at its base height. The plane was then subdivided several times, turning the plane into a dense mesh of polygons. The higher the density, the more complexity could be coaxed from the plane. The mesh was then sculpted into a stone wall by pushing and pulling polygons like clay. The material assigned to the plane changed color according to where the model’s height was raised or lowered. All work was done in the middle of the canvas with the edges left blank. The normal map was exported as an image once the middle was finished, taken into Photoshop and a function called offset applied. This function shifted the image a designated amount of pixels left or right, up or down; moving the image until its blank edges were shifted to the middle of the canvas. Finished sections of the image were used to fill in the gaps creating a seamless normal map. Once the normal map was finished, a copy of the map was created and converted to grayscale for a texture base. Color was glazed over the grayscale image, building layers of grime. Testing was done throughout the painting process making sure the texture was neither too dark nor light and was appropriate to what it was applied to.
Once the house was built, textured and in game, it was time to move on to the rest of the models (see Figure 16).

**Figure 16.** Overview of production process.

Prop models of varying degrees of complexity had to be carefully considered before, during and after they were made. Some of these props were curtains, furniture, boxes, counters, cups, and windows (see Appendix B for the model list). Windows were created to be modular so they could be placed anywhere on the house, out of concern for the house’s polygon count. It was believed that splitting the walls with window holes
would push polygon counts too high. In retrospect buildings were well under polygon limits, however it meant the windows were allowed their own set of textures separate from the house.

Large models were textured with small textures by splitting the model into sections, therefore allowing the engine to process the environment faster. *TES: Oblivion’s* models are an excellent example of this modeling and texturing method. When examining Bethesda's building models to learn how they were constructed and textured, I noticed that most of the walls were split into sections with wood studs. At first, this was dismissed as building style, fantasy architecture typically being constructed to look medieval. During the course of modeling and texturing, it was concluded that this type of architecture was a way of keeping texture sizes small. If polygons never exceeded a certain size, they could all be textured with smaller, lower file size textures.

Model polygon counts were kept intentionally low because of the modeling rule of thumb to keep polygon counts to a minimum. A survey of Bethesda’s models demonstrated that polygon count concerns were unnecessary. Most of the architectural models in *TES: Oblivion*, even simple ones, were around 6,000 polygons. In contrast, the house built for this project was around 2,500 polygons.

Summarizing the creation of a digital environment also, unintentionally, simplifies it. There was truly no linear path taken in order to bring this aspect of the project to the point where it could be deemed ‘done.’ In fact, there is still a great deal that could be done to bring the environment's overall quality up several notches. However, due to time constraints, it was better to let those details go and make note of them for
another time. In the interest of trying to explain how the digital environment came together, an outline of the process has been provided below:

- A small 'world' was created in the TES: Construction Set.
  - Test in game
- Grass was painted on the landscape.
- Ground is altered to break up height.
  - Test in game
- Region editor used to place flora and forest detritus.
  - Test in game
- Focal point models (house/barn) were placed.
- Rest of environment models placed in relation to house and barn.
  - Test in game
- Interior cells created for inside of real world house (basement, attic, first and second floors), and barn.
  - Test in game
- Create interior spaces, insert interior models.
  - Test in game
- Start preliminary lighting.
  - Test in game
- Add doors to teleport the player between exterior and interior.
  - Test in game
• Begin populating cells with custom interior props.
  o Test in game

• Adjust lighting as props were added.
  o Test in game

• Create mirror world from copy of real world.
  o Test in game

• In mirror world, replace real world models with mirror world models.
  o Test in game

• Re-light mirror world to change feel of environment.
  o Test in game

Features Cut from the Final Product

There were several features planned during the initial development of the example digital environment that were quickly cut. These features were planned when the digital environment was still being considered a small game. Due to lack of information available about modifying these features, time constraints and inexperience with programming, these features were removed. However, these features still bear mentioning, since they were part of the development process even though they did not make it into the final product.

Adding the ability to pick up objects and be affected by gravity was abandoned after a few tests. There were few tutorials available on how to add this functionality in a reliable manner. Editing the weather in *TES: Oblivion* was easy to execute, but in game, the settings did not seem to take hold. Since weather was not a
feature key to the digital environment, it was dropped. While creating the digital environment, elements of game design were considered, even though the environment was not a game itself. Placement of objects was very important because objects lead the player, telling a story along the way. Attempting to lead the player in this project would have required twice as many models as well as scripted triggers and audio cues. These would have been used to help present the original story planned for the environment. However, there was barely enough time to finish both environments, let alone triggers.

Quality control was the most time consuming aspect of creating a digital environment. Every time a new object was added to the environment, it was immediately tested. Models had to be presentable from all sides, since players were free to examine models from any angle. Textures had to be crisp from both far away and up close, because players could see them from any distance as well. Additionally, the environment had to be functional because trying to interact with a broken digital environment is frustrating.

If this project were to be done again models would be rebuilt observing modern game standards, increasing polygon counts to reflect the capabilities of the engine. The number of models created would be doubled. Textures would still be kept to the 512x512 pixel range, but more 1024x1024 pixel sized textures would be used. Lighting would be worked with more, to show off both textures and models better than what was settled for. Two more areas would be added; the exterior of the Mirror World and the interior of the Mirror World's well (see Appendix A). Decals, scripted triggers, and audio cues would be used to break up textures as well as tell the environment's story.
CHAPTER V

RECOMMENDATIONS AND
CONCLUSIONS

Introduction

In this project, methods of texturing were researched and compared to find efficient ways of producing high quality textures and normal maps for digital games. Several methods of creating textures and normal maps were examined throughout the course of this project. These methods were:

- Textures created at intended size vs. those created large and resized.
- Tiled textures of different sizes applied to different shaped models.
- UV textures of different sizes applied to different shaped models.
- Normal map creation in four different programs.

All of these texturing methods were performed in commercially available, industry standard programs which can be acquired by the public. These programs were:

- Adobe Photoshop CS2
- ZBrush v.2
- Mudbox v.2
- Silo v.2
- TES: Construction Set and TES: Oblivion
In order to make sure that testing was not done in a vacuum, a survey instrument was created and presented to the public to be compared to my observations. Results gleaned from the survey, however, were not conclusive and required further testing and refinement.

A digital environment of my own design was used as a means of applying what was learned throughout the course of the testing phase. This digital environment was developed with a variety of media as inspiration, ultimately cumulating in a simple story from which models and textures could be drawn.

The majority of the models and textures seen in this digital environment were personally developed. All of the models made for this project were created with custom textures as well. Over sixty models were created for the digital environment, with around one hundred and twenty textures generated. The only pre-existing content used for this project was foliage, cobwebs, a ladder, and doors that were retextured to fit with the rest of the custom textures.

Based on personal observation and survey feedback, it was discovered that creating textures at their intended size produced higher quality results. Painting at intended size was faster because textures did not need extra work once they were finished to ensure they were still of high quality. Intended size textures were also sharper, as rescaling always produced blurry results.

When working with tiling textures using 512x512 pixel and 1024x1024 pixel sized textures produced the highest visual quality in-game both from personal observation and audience feedback. 1024x1024 pixel textures were an acceptable compromise in terms of texture quality and file size, especially for tiling textures. Tiling
textures were effective when applied to box shapes, but not rounded or organic shapes. Applying tiling textures to round or organic shapes produced poor results, as the texture would bend and twist in ways that looked unnatural, bringing down visual quality. Small UV textures on large models looked blurry to the point of being unrecognizable. In order to be comparable to tiled textures UV textures needed to be between 1024x1024 pixels and 2048x2048 pixels, but only for large objects. Small UV textures could be applied to small or medium sized models with little to no loss in visual quality. If a large object really needed to be UV textured, it was best to split the model into several UV textures to maximize the amount of space each section got.

ZBrush produced the best normal map results and was far more useful for making unique objects, like characters or important props. Photoshop produced normal maps faster than any of the other programs, but the texture artist needed to know the technical side of normal mapping in order to produce useful results.

A texture artist must develop patience, an eye for detail and an ability to carefully observe of the world around them. The most important thing to keep in mind when making textures was that one never gets them right the first time, especially when first learning how to texture. Texturing tutorials are stepping stones for a beginning texture artist. They teach one how to think about the process of texturing, such that when a texture needs to be made, the artist can think critically and develop a set of steps to effectively create what they need. However, one still needs to create the texture several times before they are comfortable with it. I had to create bad generic wood textures ten or more times before my textures started to look like wood, because by then I had developed a process. While the normal map tests were very simple I believe what was done is
useful. I was able to discover which gave me the best results with the least amount of experience, which in turn gave me and idea of which program would be the most beneficial to learn.

Limitations to the Findings

Due to working with a single game engine though out the development process, there are parts of this project specific to TES: Oblivion that may not apply to other development environments. For example the way in which TES: Oblivion processes normal maps; the red and green channels in the normal map must be swapped for TES: Oblivion to read the map correctly in-game. Because of the way 3DS Max applies textures to models, all models in the digital environment were required to be UV mapped in some way. This method is significantly different from some engines that allow one to apply tiled textures though the game engine itself, like Unreal 2. The manner in which textures are displayed will change from engine to engine as well. A texture that looks visually appealing in one engine may not look the same in another. Model construction could be considered TES: Oblivion specific as well because of the .nif file format that was required in order to insert objects into the game. To convert objects into working .nif files with collision meshes 3DS Max had to be used. Importing objects into the game was a unique process specific to TES: Oblivion, as explained in Chapter IV.

Findings from the survey instrument were limited due to the nature of the survey. It was too long, the format was confusing, and the questions asked of participants needed to be simpler. A wider spectrum of participants would also have helped, so that
comparisons such as gamers vs. non-gamers and industry professionals vs. laypeople could be made

Recommendations

The best place to go for current information is the Internet. Websites are more up-to-date and in some cases provide an outlet to ask questions directly of an artist or groups of artists. Forums are very useful because they house archives of previous discussions to search and current discussions to take part in. Be aware that the Internet also requires one to read though a great deal of misinformation on the way to anything useful. Favorite tutorials should be saved to a hard drive and archived, because one never knows when an artist is going to take down their site. Books, while a good starting point, are outdated by the time they hit the shelf. It is acceptable to start with a book like *3D Game Art* but in order to go beyond the basics one must look for more varied sources.

A texture artist should know how color and light affects a scene. Having some foundation in traditional art media will help a texture artist more effectively develop textures to bring out the mood of the environment. A texture artist does not absolutely need a background in fine art, but it will make texturing a great deal easier. Knowing ones way around a camera helps as well; both as a way of studying light and as a way of gathering more texture resources.

Critical examination of games (especially those with construction sets) is a way of staying abreast current technologies. Stare at walls in a game, wander around, and mentally pick the game apart. In the game’s editor look at how the game is put together. Examine textures, think about how they could have been made, and recreate them. Make
up tests and carry them out in a game engine. Experiment with different texturing
techniques or ways of generating normal maps and see how they look. Finally, never stop
learning about what other people are doing. Keep reading tutorials, but pull bits and
pieces from them to adapt a personal style.

General Guidelines to Texturing for Video Games

Please remember that these are suggestions. As engines change, so will
suggestions for texture sizes, techniques, etc. However there are general guidelines for
using Photoshop, archiving and best practices for keeping oneself organized. All of these
are demonstrated in the tutorial Appendices D, E, and F.

Non-Destructive Photoshop Texturing Practices

Do Not Flatten Layers

The single most important Photoshop practice that one should pick up, in
addition to constantly saving, is *not* flattening layers in Photoshop. If one flattens their
Photoshop layers after finishing a texture, they cannot go back and further alter the
texture if needed.

Name Layers

Giving layers meaningful names saves on frustration later in the texturing
process, as one can find what they are looking for at a glance. Nothing is worse than
going back though a texture to change a single layer and having to shift though eighty
layers of ‘grunge1’, ‘grunge copy1’, ‘grunge copy copy1’.
Photoshop Folders

In the layers menu there is a button shaped like a folder, this will create a folder that one can put layers into. Give these folders meaningful names and keep parts of your texture separated with them. When I am working on a tiling texture, I keep the base texture layers separated from the grunge layers with folders. This way if I want to use the base for something else I can just copy that folder to a new file. When working on UV textures, different texture materials are separated by folders.

Layer Masks

Layer masks are a non-destructive way to hide parts of a layer. Using layer masks preserves the integrity of a layer, because the only thing worse than shifting though 80 layers named ‘grunge copy copy copy’, is realizing that parts of a layer have been erased and now have to be filled in with the clone stamp tool. Layer masks not only save the image, they also allow layer styles to show through the mask as if one actually was erasing parts of it.

Grunge Layers

Nothing in the real world is perfectly clean; there is always a hint of discoloration caused by the object’s environment. Adding that grunge to digital textures is important because it adds a level of realism to a digital object. Grunge can be anything that adds surface detail and makes a pristine texture look dirty. This can be anything from stains, to burns, water damage, or rust. Grunge can be made in a variety of ways: brushes, overlay layers, photographs, etc. The key to making good grunge is being subtle. Grunge is not the focus point of the texture; it is icing.
**Painting with Brushes**

Brushes are a great way to build up a lot of texture very quickly. Make liberal use of the brushes window. Change settings often to see what they do to different brushes; play with the scatter settings and generally just experiment. In addition, save those brushes to be used later.

You can make your own brushes in Photoshop, or find collections of brushes online. Experimenting with lots of different types of brushes is a good idea because it makes textures look more organic and less digitally generated.

Any tool in Photoshop that uses brushes can be changed; this includes the clone stamp, blur, and eraser tools. Adding some variety to the way these tools are applied will add a more organic feel to a texture.

**Learn Photoshop Hotkeys**

Knowing the Photoshop hotkeys will effectively double productivity, because you can use two hands more effectively than just one.

**Archiving Textures**

**External Archives**

Maintaining an archive separate from a main workstation is important, because sometimes computers have meltdowns, and if that work is not backed up, it is gone forever. Archives should include textures one has made and textures one has collected from various sources. Establish naming conventions early on and stick to them, it makes the archive faster to search. In addition, textures should be labeled descriptive
names which will give you a better idea of what they are. Names like ‘StainedConcrete512’ will tell you what kind of concrete and what size the texture is.

**Photographs and Copyright**

Feel free to use photos when texturing, but if you are getting them from the Internet make sure they are under Creative Commons, free, or that you have taken the image yourself. It is illegal to use anything under copyright, no matter how much you change it. If you are unsure about copyright law, look it up.

**Reference Photographs**

Always have several pieces of photo reference when texturing. Look at them often, and when in doubt find more. Make sure to save those references in your archive as well.

**Findings from Texturing Tests**

**Avoid Resizing Textures**

Always work at intended size, unless explicitly told not to. Working at intended size keeps textures from becoming pixilated and is faster than creating a texture large and editing it after it has been resized.

**Texture Sizes**

Work at 512x512 pixels or 1024x1024 pixels, especially when making tiling textures. Keep in mind that these are sizes that work well for *TES: Oblivion*. It will change from engine to engine.
Testing Textures

Textures should always be tested on the model they are intended for while they are being made, that way the texture is the correct size before it is finished and resizing can be avoided.

Conclusion

Learning to texture effectively and efficiently is difficult. One technique may work for one person, but that same technique may not work for another. Each artist works and learns differently; however, artistic diversity is what makes each artist unique. If every texture artist worked the same way games would look boring. A texture artist must be aware that techniques will change over time. Techniques change so rapidly that by the time one learns one skill set, they need to start learning another. The texture artist must be constantly learning and practicing in order to stay relevant. Much like professional fine artists, game artists must always be producing work. Being flexible and able to learn quickly will keep one productive and producing great work. Be prepared to spend a lot of time making bad textures; one in fifty is going to be useable when first starting out. Do not be afraid or discouraged when it is necessary to throw work out. Remember, the next texture will always be better.
REFERENCES


78


http://www.poopinmymouth.com

http://www.mobygames.com/game/7th-guest

http://www.mobygames.com/game-group/silent-hill-series

http://www.nevercenter.com/


http://dictionary.reference.com/browse/photoshop

http://www.pixologic.com/home.php

http://www.scifi.com/ghosthunters

http://cs.elderscrolls.com/constwiki/index.php/Main_Page

http://www.uesp.net/wiki/Tes4Mod_Talk:Modding


STORYBOARD

Storyboard 1

82
The events that happened here make the plot perfect! Sounds like a Disney story.

The perfect place for me to start my career as an up and coming urban explorer.

I'll have to make sure I take copies of my photos to Mike Lachmichael later.

Hopefully these aren'tcobwebs.
Storyboard 3

Guess I'll just wander around but see what's here first

Since the front door's already open

And a good night too

The back door's conceals no light

Shuttered all night

Good thing I bought a flashlight
Storyboard 4
Storyboard 6
Storyboard 7
Storyboard 12
Storyboard 14
Storyboard 21
Storyboard 24

1. A character is standing in a room, holding a key.
2. Another character is standing nearby, holding a key.
3. The first character says, "Not another key!"
4. The second character says, "Another key?"
5. The first character says, "Now I don't know whether the obstacles talking about this key or the other key!"
6. The second character says, "Course, I could always use a key!

Note: The text is in a handwriting style and may not be fully legible.
Storyboard 32
Not that I actually think one will be up here.

I better check the attic.
I'm gonna need a cordless one buzz off.
Storyboard 43
Storyboard 44
APPENDIX B
## MODEL AND TEXTURE LIST

<table>
<thead>
<tr>
<th>Models</th>
<th>Textures Real</th>
<th>Textures Mirror</th>
<th>Can Reuse Texture?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barn door</td>
<td>1</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>Barn exterior</td>
<td>3</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Barn interior</td>
<td>4</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Barn window exterior</td>
<td>1</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Barn window interior</td>
<td>1</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Basement Door</td>
<td>2</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>Bathroom sink</td>
<td>2</td>
<td>2</td>
<td>n</td>
</tr>
<tr>
<td>Bathtub</td>
<td>1</td>
<td>1</td>
<td>n</td>
</tr>
<tr>
<td>Bed</td>
<td>0</td>
<td>2</td>
<td>n</td>
</tr>
<tr>
<td>Bookshelf</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Cardboard box</td>
<td>1</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Ceiling light</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Coffee mug</td>
<td>0</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Couch</td>
<td>1</td>
<td>1</td>
<td>n</td>
</tr>
<tr>
<td>Curtains - narrow</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Curtains - wide</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Desk</td>
<td>0</td>
<td>3</td>
<td>y</td>
</tr>
<tr>
<td>Dining room chair</td>
<td>0</td>
<td>3</td>
<td>y</td>
</tr>
<tr>
<td>Dining room table</td>
<td>0</td>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>Dresser</td>
<td>0</td>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>Drinking glass - short</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Drinking glass - tall</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Dryer</td>
<td>1</td>
<td>1</td>
<td>y</td>
</tr>
<tr>
<td>Fireplace</td>
<td>2</td>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>Glass Mason jar - short</td>
<td>1</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Glass Mason jar - tall</td>
<td>1</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>Hand Decals X 5</td>
<td>5</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>House exterior</td>
<td>6</td>
<td>0</td>
<td>y</td>
</tr>
<tr>
<td>House exterior window - kitchen</td>
<td>2</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>House exterior window - narrow</td>
<td>2</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>House exterior window - wide</td>
<td>2</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>House exterior window -stained glass</td>
<td>1</td>
<td>0</td>
<td>n</td>
</tr>
<tr>
<td>House interior - attic</td>
<td>2</td>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>House interior - basement</td>
<td>4</td>
<td>4</td>
<td>y</td>
</tr>
<tr>
<td>House interior - first &amp; second stories</td>
<td>4</td>
<td>4</td>
<td>y</td>
</tr>
<tr>
<td>House interior window - kitchen</td>
<td>2</td>
<td>2</td>
<td>y</td>
</tr>
<tr>
<td>House interior window - narrow</td>
<td>2</td>
<td>2</td>
<td>n</td>
</tr>
<tr>
<td>House interior window - wide</td>
<td>2</td>
<td>2</td>
<td>n</td>
</tr>
<tr>
<td>Kitchen sink (in long tiled counter)</td>
<td>3</td>
<td>3</td>
<td>n</td>
</tr>
</tbody>
</table>
## Model and Texture List (Continued)

<table>
<thead>
<tr>
<th>Models</th>
<th>Textures</th>
<th>Can Reuse Texture?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real</td>
<td>Mirror</td>
</tr>
<tr>
<td>Mattress</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Painting</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Pantry shelves</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Pillow</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Plant stand</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Refrigerator</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Round rug</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sideboard cabinet</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Sideboard table</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Small table</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Stained glass hanging lamp</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Stained glass wall sconces</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Stone wall bend piece</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Stone wall pillar</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Stone wall straight piece</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Terracotta pot - large</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Terracotta pot - small</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Trunk</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Wardrobe</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Washer</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Water heater</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Well</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

86 71
APPENDIX C
SURVEY EXAMPLES

Welcome!

The following survey is being conducted as part of a research project for completion of an Interdisciplinary Studies Master's Degree in Applied Computer Graphics at California State University Chico.

Participation in this survey is 100% voluntary. There is no penalty for non-participation, and subjects can withdraw from the survey at any time by closing the browser. There are also no known benefits to the participants for completion of the survey.

Participants will remain completely anonymous; they will never be asked for a name or an email address; however, they will be asked a few simple background questions for demographic purposes. However, participants must be 18 years or older.

Participants are being asked to rate the results of a series of tests developed for the creation of textures for digital games. The participants are shown an image of a texture that has been applied to an object in a game, and then are asked to rate the texture based on a series of questions. While taking the survey, participants are asked to rate each image without comparing it to any previous images.

The survey will take, at most, 20 minutes. If you are interested in participating, please click the "Agree" button to proceed to the demographic questions and survey overview. If not, close the browser, and thank you for taking an interest.

Any questions, comments, or concerns can be directed to: Lindsey Anderson - unipegx@gmail.com
Demographic Questions

Age: 

Gender: male

Do you play video games: no

Are you a college student: no

If you are not a college student, what is your occupation:

Are or have you been employed in the video game industry: no

Do you have a background in art: no
The format of the survey is as follows:

After clicking the 'Start' button at the bottom of the screen, you will be shown an image of a texture that has been applied to a model in a digital game engine. Underneath the image are four questions asking you to rate the texture based on specific criteria.

Do you think the texture looks blurry?
Blurry textures look muddy and indistinct, making it hard to determine what a texture is.
1 = Very blurry  7 = Not blurry

Do you see any 'banding' in the texture?
Banding is where large stripes of color are seen in the texture.
1 = Yes, a great deal of banding  7 = No banding

Does the texture look pixelated?
Pixelation is when you can see the individual 'dots' known as pixels that the images are made of.
1 = The texture looks very pixelated  7 = The texture did not look pixelated

Is this texture attractive to you?
Do you like the way it looks? Is it pleasing to the eye?
1 = I do not like this texture at all  7 = I like this texture very much!

The last four images in the survey pertain to normal maps. Normal maps are used to make a simple object appear to be more complex. A detailed description of what normal mapping is along with the criteria for rating the normal map tests will be given before the normal map section.

Once you have entered your ratings, you move on to the next image. At the end of the survey you will be asked to submit the results, which will be sent to the survey administrator via email.
Does the texture sample look blurry?
Blurry textures look muddy and indistinct, making it hard to determine what a texture is.
1= Very blurry  7= Not blurry

Is there any banding in the texture?
Banding is when large stripes of color are seen in the texture.
1= Yes, there is a great deal of banding  7= I did not see any banding

Does the texture sample look pixelated?
Pixelation is when you can see the individual "dots" known as pixels that make up the image.
1= The texture looks very pixelated  7= The texture did not look pixelated

Is this texture attractive to you?
Do you like the way it looks? Is the texture pleasing to the eye?
1= I do not like this texture at all  7= I like this texture very much!
Normal Mapping & Test Criteria

The normal map works with the texture of an object to trick the mind of the viewer into thinking they are seeing something that isn’t there. This is a way to make simple objects appear very complex. Game makers are interested in this process because they are constantly walking the fine line between the quality of their game and what a computer can realistically run. In order to better understand what you’ll be looking at in the following images, I’ll provide a brief description of what normal mapping is.

In a 3D modeling program I made a highly detailed model of a stone wall. When I was done, I had the modeling program create an image of the wall and saved it as an image that other programs could read; the image is the normal map. From there I was able to take a copy of that image into another program, paint what looked like stone onto it, and then apply the normal map and the fake stone painting to a 3D model.

![Model, Normal map, Texture, Texture & Normal map in game]

The detailed model of the stone wall is only used to make the normal map. It is never applied in game. The normal map tells the game engine how light hits each stone, where it is dark, where it is light, and the result is a wall that looks like the highly detailed model, but is actually flat.
Normal Mapping & Test Criteria

On the next four images the same normal map as been recreated in four different ways. Now that you know what a normal map does, I am asking you to judge the effectiveness of each map based on the following criteria:

Could you clearly see where the normal map was on the model?
1= There was nothing visible on the model  7= I could easily see where the normal map was

Did the normal map look flat and painted on or did it look three dimensional?
1= Flat and painted on  7= Three dimensional

Did the normal map make the object appear to be more detailed and complex?
1= the object does not look more detailed  7= the object looks much more detailed

Do you think this looks good?
Do you like the way it looks? Does the normal map make the object look better?
1= The normal map does not make the object look good  7= The normal map makes the object look better!
Is the normal map on the cube clearly visible?
1 = I cannot see any indication of the normal map
7 = I easily see the normal map

Does the normal map look flat or does it look three dimensional?
1 = The normal map does not make the object look three dimensional
7 = The normal map looks three dimensional

Does the normal map make the object appear more detailed?
1 = The normal map doesn't make the object look more detailed
7 = The normal map makes the object appear more detailed

Do you think this looks good?
1 = The normal map does not make the object look good
7 = The normal map makes the object look better!
This is the end of the survey

Thank you for participating in this survey!

Your results will be emailed to the administrator of the survey after you have hit the reply button at the bottom of this screen.

The results of this survey are very valuable in determining what the general public finds to be an acceptable level of quality for textures in digital games. In turn this will help me determine the processes that will be used to create textures for the final portion of my Master’s project: Creating an environment in a commercial game engine using the techniques I have learned and developed. I greatly appreciate your time and welcome anyone to contact me if they wish.

Any questions about the nature of this project, or the survey please email:

Lindsey Anderson - unipepx@gmail.com

Thesis Committee Chair: Professor Clarke Steinbeck Ph.D, CSU Chico - ranger@ecst.csuchico.edu
APPENDIX D
WOOD TUTORIAL

Texturing Wood

Wood is a difficult texture to get right. Part of the problem with wood is that people see it so often they have preconceived notions of what looks ‘right’. For this reason, most people when texturing wood will default automatically to photographs, since creating wood grain by hand is tough and time consuming. Instead, the key to creating convincing wood is keeping it subtle and slightly on the desaturated side. This tutorial will detail my processes used to create wood from start to finish, with detailed instructions about layers, layer effects, and brushes. While this tutorial is geared towards those who use Photoshop, it could be adapted to other programs if one is more familiar with them and can apply the techniques. As with all texturing, one should have numerous reference images, and be aware of what type of wood they are creating. The texture artist should be asking themselves what species of tree this wood represents, how weathered this object is (wood fades over time), and what the object itself is.

Base Layer

- Open a new Photoshop document.
- Set canvas size and DPI.
- Name background layer “base.”
- Save document.
Open up Photoshop and create a new file. It is suggested that one make *liberal* use of the ‘CTRL + S’ save function while working. This tutorial uses a 512X512 pixel sized canvas with a 150 DPI, because this texture would be intended for next-generation games. On the background layer, drop in a base brownish color.

- Filter > Noise > Add Noise.
- Amount - around 10%.
- Gaussian checked.
- Monochromatic checked.

To the base layer apply a Noise filter. Increasing the noise amount creates more contrast in the base wood grain. New woods, or well taken care of wood will have more contrast in the base wood grain. Old weathered woods are lacking in contrast, if the wood needs to look aged the contrast should be low.
• Filter -> Blur -> Motion Blur.

• Vertical angle (up/down).

• Blur amount: 25.

Now that the base layer is suitably noisy, apply a Motion Blur to the layer. Align the blur to apply at a vertical angle. It is easier to work on an up and down texture than a side to side one.

The distance of the pixel blur depends on the size of the texture. Small textures need less distance larger textures need more. For a 512 sized canvas Motion blur is set to 25, for 1024 sized canvases it is set around 50. The resulting image looks like clear wood grain, but we want to make it more interesting by adding in whorls and maybe some knots. Painting all that by hand is time consuming though, and unless one wants to spend hours, it will not look good. Instead, there’s a filter that will do that for us.
Using Brushes to add detail

- Create a new layer.
- Choose the brush tool.
- Open the brushes window.
- Window -> Brushes (or F5).
- Set layer blending mode to Multiply or Overlay.

First, make sure that Photoshop's "wet media brushes" are loaded. If they’re not, click on the little arrow on the right in the Brushes window, go down to the bottom of the brushes list and select the very last one, Wet Media Brushes. If the brushes are not already loaded append them to the list. Appending puts the brushes at the bottom of the brush presets list.
At the bottom of the wet media brushes is “Watercolor textured surface”; select it. This special brush mimics a real media watercolor brush. It has a translucent quality which allows whatever is under to be seen through the brush strokes. Keeping in mind the base color of the wood, choose a darker complimentary color to use for the whorls. When choosing this color make sure it is not a brown that clashes with the base color. If it is too different (like using a red-brown on a yellow-brown base) it will look out of place. Set brush opacity to around 50%, again so that we can clearly see layers beneath. Paint a few horizontal lines on the new layer. Do not fill the whole canvas with lines. We want some empty space so there is separation between the whorls. Make sure to overlap lines while painting, the dark areas help add interest and variety to the whorls. There is an element of unpredictability in the way the filter is applied which adds some realism to the texture.
Remember that the layer these lines are on should also be set to either Multiply or Overlay so that it blends into the base wood grain.

Applying the Wave filter

- Filters -> Distort -> Wave.

- Generators - Anything under 50, any higher and strange distortions begin appearing in the texture.

- Wavelength - Minimum 20 or lower, Maximum 85 or lower (though not lower than the minimum). Wavelength controls the left to right of the wave. Increasing the wave length, the longer the wave will be, decreasing it shortens the wavelength creating more waves horizontally.
- Amplitude - Minimum 15 or lower, Maximum 120 or lower (again, no lower than the Minimum). Amplitude controls the length of the wave from top to bottom. The higher it is set, the more stretched out the wave will be vertically.
- Scale – Scale does not affect the image enough that it needs a setting.
- Randomize - Will generate a new version of the whorls based on the slider values.
- Undefined areas – Either "Repeat edge pixels" or “Wrap around” can be used, though “Wrap around” tends to create strange distortion along the edges that is hard to edit out.

Adjusting the Wave filter is where the most time will be spent. Slight changes in any of the variables can produce drastic changes in how the filter applies itself. Generally staying under or around 100 in any of the settings is a good rule of thumb. Going any higher can produce unwanted distortions in how the image is generated like the ones seen here on the left and right sides. We’re going to leave them for now though, because I like what’s going on in the middle and the parts where there is distortion will be removed later.
Creating Wood Planks

- Select the line tool, set to between 2-3 pixels wide.
- Create 10-11 lines representing spaces between boards.
- Apply a Noise filter set to 10%.
- If intended for a game model, test texture for size.

Hide the whorls layer for the moment by clicking on the eye icon next to the layer in the layers window. On a new layer using pure black and the line tool, draw in the spaces between the boards. For this example, I’m going to use 10 lines to make 11 boards. (Keep in mind that you are going to have to test your texture to make sure it’s not too big when it’s projected on your model.) Next, apply a noise filter to your lines. 10% or so is enough. If this texture is ultimately intended for use in a game, apply the texture to a model to check it for size in game. If it is too big or small, adjust it now.
To give the boards some depth we’ll apply some layer effects to the lines. Right click on the lines layer, and bring up the layer blending options dialogue box.

- Check Bevel and Emboss, and activate the options for it.
- Change the Style to Emboss, the technique to Smooth.
- Adjust the Depth to heighten the illusion that the grooves have some dimension.
The size and soften sliders you need to adjust according to the size of your texture. You’ll have to tweak them a bit to make them look good on what you’ve made. I leave the angle of the texture at 120 degrees, and the altitude at 30 degrees. For the gloss contour I change it to the inverted cone (the third from the left on the top row) so that the grooves look like they are going down. The Screen value is changed to a light brown to match my wood, and the black Multiply is either left alone or changed do a dark color complement. With the settings above the lines will look like this.
• Unhide whorls.

• Remove undesirable sections of whorls.

• Create new layer of whorls, removing undesirable sections.

• Repeat until all planks are filled.

Unhide the whorls layer to see how the image is shaping up. It looks pretty good at the moment, but it’s obvious that the whorls are one wave. A real wood fence/deck/wall wouldn’t look like that. The planks come from different trees, are cut differently, and may sometimes be nailed upside down. In addition the distortions on the left and right sides are distracting and need to be removed. To simulate this, we’re going to cut out parts of the wave, flip some pieces upside down, and add new layers of waves to break the pattern up.
Most of the first layer was cut out, in part because it had little variation to the texture. Now we’re going to make several layers of whorls and selectively cut out the best parts of them to make wood planks.
Make a new layer of lines. I also added a knot, just for something different.

Apply the filter again, adjusting the settings if needed.
I unhide the first layer so I could see how these whorls compared to the first set.

Again, I like what I’m seeing, and I think I might leave the overlapping whorls on the middle plank. It looks interesting.
Cut out some pieces, move planks if needed. Continue doing this until all the planks are filled. At this point, I think you’re familiar enough with the process for me to stop describing each layer. It’s going to take quite a few layers to get something that looks nice. Don’t worry if you only get one plank a layer, the more layers you have the better it will look. Just remember to use your own judgment, think about each piece, and when in doubt Google some photo reference. My finished planks look like this. Remember, yours are going to look different thanks to the wave function.
What we have at the moment looks like a nice, new, shiny hardwood floor, though the whorls are too dark and would need the opacity of their layer lowered by around 25%. Not bad, but for the sake of practice, we will apply some grunge layers so that it looks like old weathered wood.

**Weathering and Grunge**

- Create a new layer.
- Open brushes window.
- Select a scatter brush.
- Lower brush opacity.
- Set brush blending mode to Multiply.
- Set layer blending mode to Multiply.
Up until now we’ve mostly played with Photoshop’s filters. However for the next few layers, we’re going to focus a little more on the brushes window. By adjusting brush settings we can get a wide variety of effects and textures. Brushes are extremely powerful, with them you can make whole forests in a few strokes, or make wood look like it is old and nasty. Keep in mind, however, that making good looking grunge takes a lot of layers and a lot of experimentation. To start we’re going to add a base layer of dirt. We want to apply the dirt unevenly to the surface of the wood, so let’s choose a brush that has a bit of randomness to its application. Anything that has some variety to it should work fine. Remember this is just a base layer to get started on.

![Brushes Window]

Next, I go to the brush tip shape and increase the spacing to the point where each instance of the brush is just barely touching each other.
In Shape Dynamics check flips X & Y jitter, this will flip the orientation of the brush randomly as it is applied to the canvas.
In Scattering, increase the scatter along both axes, and then increase the count slightly.

![Scattering settings and sample output](image)

The result should be a brush that will do something like this when drug across the canvas. We don’t want the stroke to be that dark though, it would get too thick too fast after a couple brush strokes. Lower the opacity of the brush to 40%, and set its blending mode to Multiply.
Since I’m working with a warmish colored wood, I’m going to change my brush color to a dark green. Red and green are color complements, so when next to each other they seem more intense. Making the shadows green will make them look darker as opposed to using a dark red, or even black. With the brush at the above settings and a new layer on top of the layer stack (we want it to affect the base wood grain, the whorls and the lines) I went over the canvas a few times. It already looks a bit weathered. I’ve also changed the blending mode of the layer to Multiply, which further blends the grunge into the wood layer. At this point I’ll begin erasing parts of the grunge with the eraser tool set to scatter on a low opacity and brush it over the layer several times.
• Create a new layer.

• Change blending mode of layer to Soft Light.

• Change brush color.

• Apply Noise filter to layer (filters > Noise > Add Noise: 10%).

On a new layer with the same brush, I paint on another layer of grunge with a different color, a dark red. This time, however I am going run the Noise filter on this layer. This will help give some grit to the texture. We’re also going to set the layer blending level to Soft Light. This will blend the red into the background, but not make it so dark we can’t see what’s behind it.
• Create a new layer.

• Set blending mode to Color Dodge.

• Change brush color to a lighter color (yellow brown in this case).

• Brush in color and erase.

• Lower opacity of layer by 20%.

Now that there is an acceptable level of grunge on the wood lets add some highlights to it. On yet another layer, I set my brush to a smaller size, change the color to a more yellow brown, and add in highlights. Like with the dark colors, I first lay on a thick layer of color, erase most of it, and repeat until I’m happy with the results. The layer blending mode is also set to color dodge and reduced 20% so that the color isn’t too bright. With several passes we get something like this, it’s subtle, but the highlight helps give more depth to the surface.
Scratches

- Choose a hard edged long brush.
- Set the brush color to something light.
- Adjust brush settings.
- Apply brush, erase as needed.

Now on to scratches. For nicks and scratches, its best to use a brush with a hard edge and looks like a line. Any brush fitting these descriptions can be found in the default Photoshop brush sets. I tend to load all the brush sets Photoshop has so that I can experiment while painting. The one chosen here is particularly good for scratches, because when it’s set to a small size it looks like tiny little dents.
The amount of space between each instance of the brush is increased.
Increase the size jitter to its max and if you’re using a Wacom tablet set your control to Pen pressure. The harder you push the bigger the feathers will be. Push lightly, and the feathers will be small. Remember, scratches aren’t uniform. Also, make sure to check flip on both X and Y, we’re going for random here.

Go to the scattering tab, and increase it to 1000% across both axes.
In the Other Dynamics tab increase the opacity jitter, and set that to pen pressure as well. The harder you press, the more opaque the stroke will be.
I set the brush color to something a little more orange, it matches the colors of the wood, but should blend well as scratches. If the color ends up not looking good, I can always go back and change it.

- Apply a layer style to the scratches layer.
- Apply more brush strokes, as needed.
- Lower opacity of scratches layer, as needed.
- Add a desaturation layer to entire texture, as needed.

The blending mode of the layer is set to Soft Light to blend the scratches into the wood grain better. To give the scratches some depth I’m going to add a emboss layer style to push the brush stroke down.
The brush is applied liberally at first to get a feel for how the layer style will look when painted on with the scatter brush. What we get isn’t bad, but it is overkill.
The eraser is again set to a scatter brush and the opacity lowered so that subtle amounts of the scratches can be removed. The layer was also lowered in opacity by 20% to further blend the scratches into the wood. The result is good, but looks too saturated, so we’re going to adjust that with an adjustment layer, so that if we want to increase or decrease the saturation later we can.

![](image)

The saturation layer lowers the overall color of the entire texture by 20%, just enough to make it look more weathered and less new. The final outcome isn’t too bad, though if it was going to be a tiling texture I would have to go back and adjust it using the Offset filter and the Clone and Patch tools.
APPENDIX E
RUST TUTORIAL

Texturing Rusty Metal

When creating environmental textures an overlooked detail that adds a lot of character to an object is weathering. Weathering is the dirt, grime, or corrosion that happens to an object over the years that clues the viewer into how realistic something is. One type of weathering is corrosion or rust which occurs when a metal object has been outside. Creating convincing rust can be difficult but is a rewarding endeavor, as it adds character and age to a texture. The first step is deciding what kind of metal is being created (iron, steel, copper, etc.) and then finding appropriate references of what that metal looks like as it breaks down over time. Creating rusty metal takes quite a bit of reference before one can start, as rust can look very different given the physical location of the object, what type of object is being weathered and so on. For this tutorial a very basic sort of rust will be demonstrated, one that could be used as a base and applied to other textures to make multiple objects look aged. It is recommended for this tutorial to have all of the Photoshop brushes loaded so that one can switch between them quickly while trying to create an organic, natural looking texture. Additionally, it is recommended that the tutorial preceding this one, Texturing Wood (Appendix D), is read before beginning this one. The wood texturing tutorial goes more in-depth on adjusting brush dynamics and using layer styles, and is a good way to familiarize oneself with Photoshop. This tutorial will build more upon the use of layer styles and introduce the reader to non-destructive texturing techniques.
Base layer

- Open a new Photoshop document.
- Set canvas size and DPI.
- Name background layer “base.”
- Save document.

Create a new Photoshop document and immediately name and save it. For this tutorial, an image size of 512X512 pixels and 150 DPI was used. The larger DPI was used because this texture would ultimately be intended for use in a next generation game engine, so therefore needs a great deal of detail. It is suggested that one make liberal use of the ‘ctrl + s’ save function while working. In the layers window rename the white background layer ‘base’, this is where we will begin painting.

- Choose or create a scatter brush in the brushes window (F5).

Open the brushes window (either Window > Brushes or press F5) and choose or create a scatter brush with which to begin painting. The brushes window gives the user huge amounts of control over their brushes and is one of Photoshop’s greatest assets. In the brushes window one can adjust not only the size, shape, and spacing of a brush, but one can also add another brush on top of the primary one, textures, color variation, pen pressure effects and more. The brushes we will be using, scatter brushes, are brushes which have settings that add variation to how a brush lays down color. Increasing the amount of scatter will increase how varied a brush affects the canvas. Any brush can become a scatter brush by just clicking the check box next to ‘scatter’ in the brushes window, though some brushes work better than others with this setting. Playing with scatter is something the user should do to become more familiar with the settings.
- Change foreground color to warm red.
- Fill base layer with scatter brush.

For the base color, choose a warm brown or reddish color. Most rust is going to be on the warmer side of the color spectrum, even if the metal it is being applied to is on the cooler side. Fill the base layer with scatter brush, alternating the brush blending mode between several different styles: normal, multiply, screen, color dodge, color burn and overlay. Change the brush shape and opacity often as well. The more variation to the texture the better, though try to keep one area of the texture from standing out more than any other. This base texture, for example, needs to have the red spot in the top right edited out so it matches the rest of the image.

- Run the Texturizer filter with the ‘sandstone’ setting set to its smallest scale and a low relief.
Once the base is filled with color the Texturizer filter is run with a small scale and low relief (Filters > Texture > Texturizer). Since this is a base texture it would be acceptable to use the filter setting ‘sandstone’, because we will alter the layer further later. Normally it is better to avoid Photoshop presets because they are over used and easily spotted. But for the sake of an example, we’ll use it instead of creating a custom pattern. This is the base texture with the Texturizer set to the smallest possible scaling and a relief of 3.

- Run the Offset filter with sliders set to half the canvas size.

  Next, the Offset filter is used. Set the horizontal and vertical values to half the size of the texture, so that the edges are now in the middle of them image (Filters > Other > Offset). This filter will wrap the texture around the canvas, bringing the edges to the middle.
With the edges now in the middle of the image they can easily be blended into the rest of the texture. While editing the edges, the texture can also be made to look more uniform. For example, the red spot that was in the upper right is now in the lower left, and can be removed at the same time as the lines.

- Use the clone stamp and patch tools to blend edges into image.
Using the clone stamp and the patch tool, we remove the edges bisecting the image so the texture looks seamless again. While using these tools, be careful to avoid too many repeating patterns or blurred pixels, which are distracting. The base layer is actually a little sloppy for my taste, but again, for the sake of demonstration it works well. Notice that, over all, the image is the same tone. If one were to squint at it, blurring the details, there is no part of the texture that stands out more than any other part.

At this point, the texture would be tested for size in an engine to make sure that fit the proportions of the object it would be applied to.

Layer Two

- Create a new layer above the base layer; either copy and paste base layer or repeat previous steps to make an entirely new image.
Now that the base is done and tiling, one can either copy and paste it onto a new layer or repeat the previous steps to make another layer. If copying the base layer, it should be lightened and rotated 90 or 180 degrees. Instead of copying the base layer, I prefer to paint an entirely new image so that there is less chance of it looking too similar to the base. The image below is a new painted layer, Texturizer and Offset so it tiles. It is lighter than the base layer so that it does not get lost or look like it is blending in to the layer below it. It may need further color correction, however, once parts of it are pulled away to see the base.

- Apply a layer mask to top layer.
- Paint layer mask with a hard edged brush to create rough edged holes.
At this point, a layer mask is applied to the top layer. The layer mask is a black and white image that hides parts of a layer so that what is underneath can be seen. The mask is the white box to the right of the top layer in the image below. The chain between the two boxes tells us that the mask is linked to the layer. To create a new layer mask one simply selects the layer they want the mask applied to and click the button at the bottom of the layers menu with the circle inside the square.

To paint on the layer mask the preview image to the right must be activated, this is done by clicking on the layer mask preview image. The layer is activated when it is outlined by black brackets. When the mask is activated, the brush tool colors will change to black and white, indicating that you can begin painting the mask. Black represents transparent while white represents opaque. The brush functions just like it would if one was laying down color. Likewise, the brush shape can be changed and altered in the brushes window. For rust, the brush should be hard edged, because rust flakes off in hard edged chunks. The image below shows a layer with a layer mask applied, and how that looks in the layers window.
With the layer mask applied, this is how the texture now looks. The darker colors are the base texture while the lighter slightly desaturated colors are the layer on top.

- Double click on the layer to add a layer style.
- Adjust layer style settings to add height and depth to top layer.
Now that there is a layer mask applied, we can add a layer style as well. Layer styles alter the look of a layer, adding special effects like drop shadows, glows, color, and texture overlays. Layer styles are a way of adding 3D looking effects without having to paint them by hand. In this case, we will be using a layer style to add a highlight and shadow to the places where we’ve painted a layer mask, making the texture look like it is a corroded surface. Apply a bevel and emboss by clicking the check box. Reduce edges to 2 with no softening, rust is hard edged and should not look blurry along the edges. Turn on anti-aliasing to help the edge look smooth. Change highlight and shadow to match texture by clicking on the color preview boxes and color pick a light orange for the highlight and a dark red or green for the shadow. Using green is a way of making the shadow very dark since red and green are color complements. Complements play off each other by intensifying the hue when placed next to each other.

With these settings, the result of the layer style looks like this. One can now easily tell between the top and bottom layers. However, the edges are too soft and it does not look like rust, though it is getting close. The problem is that the edges are still too thick.
Rust flakes off in layers measured in millimeters, this looks like chunks were scooped out of the top layer.

To try and fix this a contour setting was added to the emboss setting to further adjust how the highlight and shadow were applied to the layer. All of the contour settings were cycled through to see how they altered the look of the texture. The one below was settled on.
The result of the bevel and emboss with a contour is below. This is also the texture with the offset filter run and the layer mask altered so that it tiles. The end result is good enough to pass for rust if applied to an object. However I’d go back and do more color correction on the top layer, perhaps desaturating and lightening it further to make it stand out from the bottom layer.
A texture such as this one does not have to solely be used as a tiling texture. It could also be made as a large texture and then parts removed to apply to other textures or objects. For example, if one has a pipe that needs to be rusty, parts of the large base texture could be copied and pasted onto the pipe to make rusty edges. Another way to apply rust to a large area of an object is to apply a copy of the rust image and add a layer mask so that sections of it can be seen; much the way the top layer of this layer was created, only more deliberate, so that the rust does not look arbitrary.
APPENDIX F
Concrete, like wood, is one of those textures that if it is not done right looks very out of place. Most people will default to using photographs to make concrete textures, since it’s so easy to get wrong. *3D Game Textures* includes tutorials for creating concrete that are good bases, but look flat compared to the textures seen in games today. This tutorial will be creating a very basic sort of concrete, one you’d see on the sidewalk for example, in addition to adding some large cracking for the sake of practice. Generally, concrete is made of small rocks which you can see if you look close enough. Aggregate concrete is made of larger rocks, though still not much bigger than a thumbnail, and is a style you see mostly in older walls or sidewalk. For the sake of simplicity, we’ll be doing the smaller rock concrete, because it will use less photo resources. It is recommended that the reader have read both the wood and rust texturing tutorials preceding this one, as the use of layer styles, brushes, filters and blending modes were covered in-depth. This tutorial, while building off of previously learned techniques, will be covering other Photoshop functions: namely using the Offset filter to create tiling textures, using the Texturizer filter and further instruction on non-destructive texturing techniques.

**Starting the Texture**

- Find photo reference.
- Save as a PSD file somewhere easy to navigate to.
First, find a reference image to work from, both as a visual guide for creating the texture and for later use in the Texturizer filter. Save this image as a Photoshop document (.psd) and put it somewhere easy to find. Here is a small section of the texture I found online (Smith & Adnin, 2006). This is a very rough section of concrete; it has seen a lot of weathering and has been broken up fairly severely. When it was freshly laid, the concrete would have been smooth and clean. We are going to make the clean version then add more layers to make it grungy. Using non-destructive texturing techniques will mean that the clean version of the concrete can be used in other projects.

![Concrete Texture](image)


**Base Layer**

- Open a new Photoshop document.

- Set canvas size and DPI.
• Name background layer “base.”

• Save document.

Start a new document in Photoshop. 512X512 pixels with 150 DPI. We’re using a higher DPI because we’re making a texture meant for next-generation games. It is suggested that one make liberal use of the ‘ctrl + s’ save function while working. Set the foreground color in the color pallet on the tool bar to 50% gray, leave the background color white.

• Filter > Render > Render Clouds to fill canvas.

To fill the canvas we are going to use a filter function called Render Clouds. Render clouds uses the current foreground/background colors (the ones we just set) when
rendering a random cloud field on the canvas. This filter is helpful for many different techniques, some we’ll see later on in the tutorial.

- Filter > Noise > Add Noise.
- Run Noise filter on base.

Next, we add some to give the texture some grit. Concrete is not smooth, even when it is new. The noise filter shouldn’t be set too high, 4% at most. It should be just enough to break up the smoothness of the clouds.
- Filter > Noise > Median.

- Set Median filter to 1.

To keep the base from looking too pixilated, we’ll use the Median filter set to 1 to blur the edges of the Noise filter results. The results are not much to look at right now, especially since it looks too big.
- Reduce the size of the base by half.

- Select the whole canvas (CTRL + A) and turn on the transform tool (CTRL + T) to resize.

- Duplicate and move layers so that the canvas is again filled.

Since the texture currently looks too big, we’ll reduce its size by half. This would also be a good time to test the texture if it is intended for a 3D model in a game. If the size is acceptable, it does not have to be adjusted. But for the sake of demonstration, this texture will be reduced. The halved texture is repeated in each quadrant of the canvas by copying the base layer three times and moving pixels. The resulting image looks very tiled and needs to be edited so that it looks more uniform across the entire canvas.
- Image > Adjustments > Curves.

If one was to squint at the canvas as it is currently, it would appear to be several light splotches on a middle gray background. If we were to tile this image, it would create an unwanted pattern across a large area. What we want to do first then, is lower the contrast between the light and the middle gray areas. To lower the contrast we will use a curves layer which gives us more control over lightening the dark areas and darkening the light areas.
The resulting image is still too bright in the light areas, but those will be removed though other means, namely liberal use of the Patch and Clone Stamp tools.
After attacking the texture with the Patch tool and Clone Stamp tool the texture looks less like it would create a distinct pattern. Additionally, if squinting at it the texture has a more even tone with no areas jumping out at the viewer immediately.

- Filter > Other > Offset.

The Offset filter is a staple of tiling texture creation. The filter allows the user to shift their texture up or down, left or right a set number of pixels in order to edit edges more easily. The Horizontal and Vertical shifts are both set to half the size of the canvas, so that the edges are moved to the middle, making them much more accessible.
It can barely be seen here, but there are two lines quartering the canvas, one horizontal and one vertical cutting the texture into quadrants. Those lines would be removed.

- Filter > Texture > Texturizer.
- Load a custom Texture.
- Set Scaling to a very low value.
- Set relief to 2 or 3.
The Texturizer filter is often abused in Photoshop. The default settings are over used, and to those who are familiar with the program, easy to recognize. This is unfortunate because the Texturizer filter can add a great deal of character to a texture. Lucky for us there is a way to utilize it without resorting to the defaults. Navigate to this window and next to the drop down where it says 'texture' click on the little arrow to the right. Select ‘Load texture’, and navigate to where the reference image was saved as a PSD. Since mine was a fairly light image, I increased the relief quite a bit.

![Texturizer Filter Window](image)

The resulting image is well on its way to looking like a clean concrete. Now we just need to do some touch up to further it along.
Run the Offset filter again and edit the edges to insure that it tiles.
At this point the texture should be tested on an object in game to see how it looks scale wise. I know that I tend to make my textures about half as big as they should be, so I’m going to go ahead and reduce this texture by a half and refill the canvas with it. I can see lines in the texture that I’m going to have to edit out.

With the lines edited out I’ll run the Offset filter again checking my edges to make sure that they tile properly. When making tiling textures it’s a good habit to run the Offset filter often, to keep checking that the texture is tiling smoothly. The texture looks like clean freshly laid concrete, though some of the darker spots should be evened out so that it would tile with less of a pattern.
At this point, we’ve got a clean base texture that could work well on its own or as a base for another texture. If it was going to stand alone it would need to be tested and edited further to make sure that it did not tile with a pattern (which it does). For the purposes of this tutorial, we are going to move on to adding weathering and grunge to give the concrete some age.

**Weathering & Grunge**

- Duplicate base layer.
- Rotate 180 degrees.
- Invert color (CTRL + I).
- Move dark layer to bottom of layer stack.
As a way of keeping the texture consistent, we copy the base layer and rotate it 180 degrees. Viewers will not notice the very slight variation in texture difference, since they won’t be seeing the entirety of both textures at the same time. We invert the layer colors to make sure that the highlights and shadows which were previously made with the Texturizer filter remain consistent (highlight from the top).

While this makes the layer darker, it just means that this new layer is moved to the bottom of the layer stack while the original base is on top. We do this because in the reference image from the beginning of the tutorial we see the dark parts of the concrete are where chunks have been removed, while the light parts are where the original concrete is still present and therefore slightly elevated.

- Create a new layer (on top of light layer).
- Set foreground and background colors to default (black and white).
- Filter > Render > Render Clouds.
- Filter > Render > Difference Clouds.
To create those large islands of missing chunks we’re going to again use the Render Clouds filter. Set the foreground and background colors of the paint brush to default (black and white), create a new layer and apply the Render Clouds filter. As soon as the filter has filled the canvas immediately run the filter Difference Clouds. This reapplies the clouds filter, inversing what is currently on the canvas. In this case it makes the image dark.

- Image > Adjustments > Levels.

Now open up the levels adjustment window and alter the image so that the black and white sections are hard edged. This means moving the light color slider (far right) farther into the dark spectrum, while at the same time setting the medium slider close to the dark slider. The screen shot below demonstrates what I mean.
What this does is increase the contrast of the image so that there is a hard-edged line between the light and dark portions of the image.

At this point the layers window should look something like this: the new layer (layer 1 in this case) is the layer we just created, underneath it is the light colored layer
concrete, while the dark layer on the bottom. Using the layer we just made we’re going to create those large cracks we saw in the reference image.

- Apply layer mask to light layer (middle).
- Select the Magic Wand tool.
- Activate clouds layer and select dark area with wand tool.
- Activate layer mask of light layer.
- Select Paint Bucket tool & flood fill layer mask with black.

Apply a layer mask to the light layer (the middle one) by clicking on the icon that has a circle in a square at the bottom of the layers window. Now activate the top layer that has the modified clouds render, and using the Magic Wand tool select the black portion of the layer. Activate the layer mask of the middle layer by clicking on it. Select the Paint Bucket tool and make sure the foreground color is set to black. Drop black into the still selected pixels on the mask.
The layers window should look like this, with the mask looking the same as the clouds layer above it. Click the eye next to the clouds layer to turn the layer off. The lighter texture should now have holes in it where the darker texture shows though.
The space between the islands of light concrete is too large for a tiling texture meant to be used across large unbroken areas. Something like this would tile in a very noticeable pattern that would be undesirable across a wall or street. For the moment we’re not going to worry about filling in gaps, instead we’re going to press on to layer styles first.

- Double click light layer to add a layer style.
- Adjust layer style to give concrete slight bevel.

A layer style is added to the middle layer to make it pop more from the bottom layer. This layer style adds a highlight and a shadow to the edges of the layer enhancing the illusion that it is removed from the darker layer.

While the concrete looks more three-dimensional now the layer needs to be filled in more to look like convincing concrete.
To fill with more of the light concrete, a second render clouds layer was made in the same manner as the previous one. The black areas were selected of the new layer were selected, the layer mask of the light layer was activated and again flood filled with more black on top of the previous selection, filling in the large spaces. The offset filter was then run on the layer mask, ensuring it would tile. We’re getting close, but the edges of the bevel are too smooth to look like broken concrete.
Using a hard edged scatter brush, I went back over the edges of the light concrete layer mask to break the smoothness of the lines. Again, while the edges are being gone over I am also using the Offset filter to insure that the texture continues to tile correctly.
We are left with what looks like clean, but broken concrete. Now it’s time to add some grunge layers to this texture to make it look like it is not brand new.

Make a new layer and select a textured brush with a lot of scatter to it. Set the brush and layer blending modes to Overlay or Multiply. Make sure the scatter brush has some texture to it and that wet edges is checked. Brush opacity should be low, no more than 50%. Offset again to clean up edges. The example below is only one layer of grunge with a low opacity. But already the concrete is starting to look dirty.
This will take quite a few layers to get to a finished weathered concrete. There’s really no way to show this brush stroke for brush stroke, unless watching a video. For the most part the creation of grunge takes a lot of painting on, erasing off then fiddling with layer settings. A lot of these layers have their opacity turned down to 50% or less with blending layers that further blend them into the surface of the texture. The best rule of thumb is to keep grunge simple and subtle. The layers set up looks like this in Photoshop:
When working on a concrete like this, the edges where the layer transitions from light to dark are going to be the selling point of the texture. In this texture, for example, the edges of the light areas have been highlighted with a color dodge layer and a brush set to a light brown. In the dark patches, a warmer brown was used to lend further contrast between the light and dark layers. The layers alternate between dark and light grunge, to help give a rich sense of texture depth.
Unfortunately, this texture would require further editing to make into a tiling texture. The concrete cracks would make a very recognizable pattern if tiled across a large surface. To make this a better tiling texture it would have to be more uniform across the surface of the texture with the cracks less distinct. This tutorial, however, demonstrates many techniques which could be used for the creation of this or many other types of textures.

REFERENCES

NORMAL MAP TUTORIAL

Generating Normal Maps in Photoshop

Using Photoshop is the speediest and simplest way to create normal maps. It’s as simple as just running a filter on an image. To create a normal map one should have NVIDIA’s normal map filter plug-in installed prior to beginning this tutorial. The filter can be found at NVIDIA’s website (2008). Creating effective normal maps in Photoshop is a slightly more complicated process, as the results one tends to get at first are less than appealing. For the purposes of this tutorial, I will demonstrate how to create a useful normal map in Photoshop using an image of a door I put together. This texture would be applied to a door model, turned into a decal or added to a wall model. For most next generation games the door frame and door knob would actually be modeled, but for the sake of demonstration we’ll be making a normal map with both of these details included. Doors are a very typical texture used in games that benefit from an accurate normal map. With both a good door texture and a good normal map a lot of unneeded polygons can be removed from a scene. The door created for this tutorial is a combination of photographic resources and painted resources.
Preparing an Image for the Normal Map Filter

The first step is to take an image and remove all the color from it. This can be done using either the desaturation adjustment or if one is using Photoshop CS3 or CS4 the black & white adjustment function. The only difference between the two functions is that the Black and White adjustment gives one more control over how color is removed from the image. One can manually adjust the sliders for each color of the image as it is converted. Since I’ll be going in and manually painting in where the darks and lights are I just use the desaturation adjustment. At this point one may be wondering why I am converting the color image to black and white. The reason for this is that the filter is looking at the overall tone of an image when it converts it into a normal map. Places where the image is dark will be converted so that they are recessed into the normal map, while light areas are popped out. This is useful for creating shadows and simulated
geometry. The filter is not intelligent, however, and it cannot tell the difference between a shadow and a dark stain. For that reason, we must edit the black and white image to make sure that it will appropriately reflect where there is and is not depth. For example, right now the door knob is extremely dark. If this image was run through the normal map filter as is, the door knob would look recessed into the door in-game. We do not want this, so we’re going to have to alter the knob so that it looks lighter.

- Select door knob and lighten with a curves adjustment
- Duplicate door layer & set blending mode to multiply
- Create a new layer, set to multiply and darken edges around door frame

To get this next image the door knob was selected and a curves adjustment applied to it to make it much lighter. Then the image was duplicated and its blending mode set to multiply so that the darks were darker and the lights were lighter. Next, the
frame of the door around the inside edges was darkened with more dark lines blurred and also set to multiply. Now the door knob is popping out and the rest of the door looks recessed.

- Create a new layer and set blending mode to Color Dodge.
- Select a large brush.
- Paint lightly with a light gray or white over entire layer.
- Erase away places where the texture must remain dark (corners, edges).

The next image is a brightening layer, color dodge is the blending mode and the whole door has been lightly brushed with a light gray to lighten the light areas. Places where the texture must remain dark are erased away, leaving the dark colors from the previous layer intact.
- Create a new curves adjustment layer.
- Increase contrast.

Now that it has been established where the light and dark need to be, a curves layer is added to the image. The end result is an image with higher contrast which will make a more distinct normal map when run through the filter. The image below shows the adjustment used.
The resulting image is of much higher contrast and will give us a better normal map.
Using the Normal Map Filter

- Save file.
- Layers > Flatten image.
- Copy flattened image into a separate file and undo flatten image.
- On new image run Filters > NVIDIA tools > Normal Map filter.

Always make sure to flatten your layers before running the filter. The program is picky sometimes when you have more than one. Do not save your image as the flattened version, if you do you cannot come back to previous layers and make alterations. Always make your normal maps in a file separate from the original document. In the normal map filter window leave the settings at default and click OK.
The resulting normal map does not look like it would produce a very strong in-game projection, so we’re going to use a few techniques to alter it.

Using Color Channels

- Open the Channels window (Window > Channels).

The first technique we’re going to adjust the normal map is by altering the color channels. Bring up the channels window in Photoshop (Window > Channels). This should be an RGB (Red, Green, and Blue) image, so there should be four channels: red, green, and blue and RGB. The red, green, and blue channels are gray scale images that can be altered like any other layer in Photoshop. For example, the image below is the red channel in Photoshop.
• Layers > Adjustments > Brightness & Contrast.

• Apply exact values of contrast to the R,G,B channels.

These three layers will be individually activated and a brightness & contrast adjustment applied to each to strengthen the overall look of the normal map. The same amount of contrast should be used every time, so that there aren’t any strange distortions in the normal map.

• Copy normal map and paste to new layer.

• Set blending mode of new layer to overlay and adjust opacity of layer as needed.
Using the adjustment can be used multiple times until the desired results are achieved. However, another way to adjust the normal is by duplicating the image itself on a new layer and changing its blending mode. For example, the image below has been duplicated and its blending mode set to Overlay, so that it strengthens the overall look of the normal map greatly.

The last thing left to do would be to test the normal map in a game engine with the texture it was made from. From there adjustments can be made back in Photoshop to either increase or decrease the strength of the normal map.

REFERENCES

IN-GAME SCREENSHOTS