THE REAL TIME ART PRODUCTION PIPELINE:
AN EXAMINATION OF MODELING FOR SURFACING AS USED BY
PROFESSIONALS FROM ELECTRONIC ARTS TIBURON

THESIS

Presented in Partial Fulfillment of the Requirements for
the Degree Masters of Fine Arts in Design in the Graduate
School of The Ohio State University

By
Joseph Leslie Harmon

The Ohio State University
2006

Thesis Committee:
Professor Paul Nini, Advisor
Professor Maria Palazzi
Professor Matt Lewis

Approved by

[Signature]
Advisor
Graduate Program in Industrial, Interior,
and Visual Communication Design
Copyright by
Joseph Leslie Harmon
2006
ABSTRACT

Developing art assets for real-time rendering requires 3D art generation techniques which tailor to rendering efficiency. Modeling for surfacing (MFS) is one of the techniques; it streamlines the processes of modeling and surfacing for real-time environment objects. MFS has not received wide exposure through published sources, both printed and web. The lack of published MFS information is concerning because it is a valuable technique used by industry professionals.

A study of real-time art production pipelines employed at the Tiburon studio of Electronic Arts was conducted to research the use of MFS within a single professional production pipeline. The research observed artists and their working process as they prepared environments for use within a video game created at Tiburon. The research was collected through direct observations by the researcher and through written surveys of artists from a game's pipeline.

The observation and survey research data collected from the art production pipeline at Tiburon showed that MFS is an effective and beneficial modeling process for artists. The research also uncovered the game development process used by Tiburon, the existence of the environment artist and technical artist, and the use of MFS by the environment artist.

The goals of this thesis are to present MFS and its use within real-time art production pipelines, to research the use of MFS within a professional real-time art production pipeline, and for the thesis to become a resource for students preparing themselves for work on a real-time art production pipeline.
ACKNOWLEDGMENTS

I would like to thank Traci Temple for mentoring me during high school, supporting me during my undergraduate work, and recommending me to replace her as a graduate assistant. My academic career could not have been a success without her.

I would like to thank Paul Nini for acting as my graduate committee advisor. Paul played a supporting role in the development of my thesis topic by allowing me to explore many options before settling on the topic of modeling for surfacing.

I would like to thank Maria Palazzi for her help with the writing of the thesis. Maria nurtured my growth as a writer by helping me with the direction and organization of the thesis. She also provided many resources from her experience in the 3D industry.

I would like to thank Matt Lewis for his insightful feedback of the thesis and the research. Matt was also very skillful in keeping my writing “humble” and made sure that I did not become too bold with any of my statements.

I would like to thank my friends at Enarson hall who provided me with an unforgettable work experience. Beth McGuirey, Allen Kraus, Preston Pickett, Karin Henn, and Kristina Emick are the most outstanding group of individuals I have ever had the privilege to work with.

Finally I would like to thank God, my parents, my brother, and my friends. Completing my masters degree would have been impossible without their love and support.
VITA

March 16, 1982 .......... Born - Columbus, Ohio
2004 ................. B.S. Visual Communication Design, The Ohio State University

FIELDS OF STUDY

Major Field: Industrial, Interior, and Visual Communication Design
Area of Emphasis: Computer Animation and Visualization
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>ii</td>
</tr>
<tr>
<td>Vita</td>
<td>iii</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
<tr>
<td>Chapters:</td>
<td></td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Modeling for Surfacing</td>
<td>7</td>
</tr>
<tr>
<td>3. Game Development at EA Tiburon</td>
<td>11</td>
</tr>
<tr>
<td>4. The Environment Artist</td>
<td>20</td>
</tr>
<tr>
<td>5. The Technical Artist</td>
<td>24</td>
</tr>
<tr>
<td>6. Survey Analysis</td>
<td>27</td>
</tr>
<tr>
<td>7. MFS Concepts and Scenarios</td>
<td>37</td>
</tr>
<tr>
<td>8. MFS Process Supposition</td>
<td>61</td>
</tr>
<tr>
<td>9. Conclusion</td>
<td>74</td>
</tr>
<tr>
<td>Appendices:</td>
<td></td>
</tr>
<tr>
<td>A. Modeling and Surfacing Terms</td>
<td>80</td>
</tr>
<tr>
<td>B. Internet Based MFS Search Study</td>
<td>85</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An example of the CG movie art production pipeline</td>
<td>4</td>
</tr>
<tr>
<td>2.1</td>
<td>Overview of the production pipeline</td>
<td>8</td>
</tr>
<tr>
<td>2.2</td>
<td>Real-time environment object modeling influences and MFS</td>
<td>10</td>
</tr>
<tr>
<td>3.1</td>
<td>The macro view of the game development process</td>
<td>12</td>
</tr>
<tr>
<td>3.2</td>
<td>The micro view of art asset development</td>
<td>12</td>
</tr>
<tr>
<td>7.1</td>
<td>MFS component connections</td>
<td>38</td>
</tr>
<tr>
<td>7.2</td>
<td>The geometry and UV map relationship</td>
<td>39</td>
</tr>
<tr>
<td>7.3</td>
<td>Resolving t-junction problems</td>
<td>45</td>
</tr>
<tr>
<td>7.4</td>
<td>Scenario one: MFS process shot one</td>
<td>47</td>
</tr>
<tr>
<td>7.5</td>
<td>Scenario one: MFS process shot two</td>
<td>47</td>
</tr>
<tr>
<td>7.6</td>
<td>Scenario one: MFS process shot three</td>
<td>48</td>
</tr>
<tr>
<td>7.7</td>
<td>Scenario one: MFS process shot four</td>
<td>49</td>
</tr>
<tr>
<td>7.8</td>
<td>Scenario one: MFS process shot five</td>
<td>50</td>
</tr>
<tr>
<td>7.9</td>
<td>Scenario one: MFS process shot six</td>
<td>51</td>
</tr>
<tr>
<td>7.10</td>
<td>Scenario one: MFS process shot seven</td>
<td>52</td>
</tr>
<tr>
<td>7.11</td>
<td>Scenario one: MFS process shot eight</td>
<td>52</td>
</tr>
<tr>
<td>7.12</td>
<td>Scenario one: MFS process shot nine</td>
<td>53</td>
</tr>
<tr>
<td>7.13</td>
<td>Scenario one: MFS process shot ten</td>
<td>54</td>
</tr>
<tr>
<td>7.14</td>
<td>Scenario two: MFS process shot one</td>
<td>55</td>
</tr>
<tr>
<td>7.15</td>
<td>Scenario two: MFS process shot two</td>
<td>55</td>
</tr>
<tr>
<td>7.16</td>
<td>Scenario two: MFS process shot three</td>
<td>56</td>
</tr>
</tbody>
</table>
7.17 Scenario two: MFS process shot four ........................................ 57
7.18 Scenario two: MFS process shot five ..................................... 57
7.19 Scenario two: Side by side comparison one ............................. 59
7.20 Scenario two: Side by side comparison two ............................. 59
7.21 Scenario two: Side by side comparison three ............................ 60
8.1 Image from The Prince of Persia: Two Thrones .......................... 62
8.2 The terrace ........................................................................... 63
8.3 The scale of the terrace ............................................................ 64
8.4 The textures used within the terrace ........................................ 65
8.5 The geometric subdivision of the terrace from the non-MFS process .......................................................... 67
8.6 The separation of geometry from the non-MFS process ............... 67
8.7 The texture file and texture layout for surfacing from the non-MFS process ............................................ 68
8.8 The geometric subdivision of the terrace from the MFS process .... 70
8.9 The separation of geometry from the MFS process ..................... 70
8.10 The texture file and texture layout for surfacing from the MFS process .................................................. 71
8.11 A comparison between the geometry of the MFS and non-MFS processes .................................................. 72
B.1 Website knowledge base search results ................................... 86
B.2 Website knowledge base search evaluation ............................... 86
CHAPTER 1

INTRODUCTION

The computer graphics special effects and feature animation industry (CG) and the real-time 3D video game industry (real-time) have nearly paralleled one another since their beginnings. At first, only CG was producing 3D art because the tools and technology were not yet suited for the development of real-time assets. However, improvements in hardware technology eventually allowed real-time to begin their own productions. Craig Hoffman, an experienced artist with over 15 years of experience in both industries, reflects on the emergence of real-time: “As 3D chips [hardware] in the [video] game consoles and on PCs began to mimic the hardware rendering chips in high end graphics workstations like SGI/Sun/etc; the same techniques (poly modeling, texturing, gouraud/phong shading) used for visualization in the design/CAD community and eventually film (at least for the interactive model and animation creation, but not final images) were translated over into [video] games.” (C. Hoffman, personal communication, November 15, 2006)

When real-time had the ability to create 3D art, the two industries began to take similar steps forward in the quality of their delivered products. Factors that helped the industries grow in the same direction were the similarities of problems in art production, the use of similar production tools, and the swapping of talent. Even though both industries were sharing information and trading talent, there was still a significant gap between the visuals delivered by CG and real-time.
Hardware limitations involved with rendering frames in real-time forced compromises in the quality of visuals.\(^2\) Some of the notable compromises due to hardware limitations were the number of renderable polygons per frame and the pixel size of textures. The gap was not crossable for real-time until the emergence of programmable GPUs and shading languages. Interestingly, Larry Gritz, one of the original developers of Renderman, pioneered the development of programmable pixel shaders for real-time.\(^3\)

The ability to use programmable shaders allowed real-time to cross the gap of visual disparity between the two industries. The programmable shaders were efficient enough to be run on real-time hardware, which opened up new development opportunities. Real-time artists now had the ability to use similar techniques and authoring tools to those used by CGI for the generation of visuals.\(^4\)

Throughout the history of real-time art production there have been many trends. Two of those trends still impact the industry today. First is the trend of real-time attempting to produce the same visuals as those of CGI. Real-time has made great improvements in visuals, but is still just behind CGI in terms of realism and fidelity. Second is the trend of real-time to adapt techniques, either those from CGI or their own, to maximize rendering efficiency. A good example of this is Gritz's adaptation of the Renderman rendering software to work with GPUs.

The second trend also affects the art production process used by real-time. At a glance both industries seem to share similar production processes. An object must be modeled before it can be surfaced or lit, so in simple terms they are similar.\(^5\) However, there are considerable differences that prevent the two industries from adopting a single unified art production process.

Kevin Noone, a technical artist with experience working in both industries says: “In [CGI] you’re rendering out a bunch of still images and that’s the final product. In games you’re creating a piece of software, and a very complicated piece of software at that. It has sound, graphics, AI, etc., has to run at optimal speeds, and never be allowed to crash.” (K.Noone, personal communication, November 14, 2006)
Limitations from real-time hardware is affecting the roles of those involved with the real-time art production process. This was observed at Electronic Arts (EA) Tiburon during research for this thesis. Observations at Tiburon showed that some specialties, like lighting, were influenced heavily by CG, where other positions were adjusted for real-time limitations.

Tiburon has adjusted the specialized positions of modeling and surfacing for environment art by combining them into a single semi-generalized position. A semi-generalized position is one that combines the knowledge and techniques of at least two specialties. This thesis explores the semi-generalized overlap between modeling and surfacing for the creation of environment art, referring to it as the "modeling for surfacing" (MFS) process.

Professionals from the real-time industry have written about MFS in relationship to the real-time art production pipeline; however, the individual writings of these authors have yet to present MFS in its entirety. Many of the writings lightly touch on the relationship between the processes of modeling and surfacing and its importance; presenting modeling and surfacing as independent specialties from one another. This is especially problematic because, after direct observation of Tiburon's production pipeline, it has become apparent that modeling and surfacing should be viewed in tandem. Therefore students preparing for a career on a real-time production pipeline should understand the impact MFS has on the asset production process. This study combines published MFS writings with the research collected during this thesis to present a complete, focused, and updated view of MFS within a real-time art production pipeline.

Audience

This thesis addresses students who are novice artists of real-time objects and environments. A novice artist is defined as an artist who has learned the basic fundamentals of modeling and surfacing, has had some experience with applied modeling and surfacing, and is continually
Figure 1.1 This is the CG movie art production pipeline as outlined by Kerlow (2004) in his book, The Art of 3D Computer Animation and Effects.
adding to their modeling and surfacing foundation. The novice artist must have experience with the modeling and surfacing concepts found in appendix A to understand and apply MFS.

The Research

A study of real-time art production pipelines employed at the Tiburon studio of Electronic Arts (EA) during the summer of 2006 was conducted to research the use of MFS within a single professional production pipeline. The researcher observed environment artists and their working process as they prepared environments for use within a video game created at Tiburon. The research was collected through direct observations by the researcher and through written surveys of environment artists (see appendix C). The research reflects the production pipeline practices of one development team at Tiburon. These practices are not necessarily universal across all development studios.

The study was organized into two agendas. The first agenda was to observe a production pipeline in use at the studio. Broad observations were made of the entire art production pipeline and then narrowed to the environment artist position within an environment art pod (team). The second agenda was to survey an environment art pod for their use and application of MFS. The results of the research are presented later in the thesis.

Goals

The goals of this thesis are to present MFS and its use within real-time art production pipelines, to research the use of MFS within a professional real-time art production pipeline, and for the thesis to become a resource for students preparing themselves for work on a real-time art production pipeline.
END NOTES

1. C. Hoffman, personal communication, November 15, 2006
2. F. Mendoza, personal communication, November 14, 2006
3. K. Noone, personal communication, November 14, 2006
4. C. Hoffman, personal communication, November 15, 2006
5. C. Hoffman, personal communication, November 15, 2006
CHAPTER 2

MODELING FOR SURFACING

The development of an art asset progresses through several specialties on an art production pipeline, starting with modeling (see figure 2.1). Art pipelines start with modeling because the other specialties (surfacing, lighting, rendering, level artwork, animation, rigging) are dependent upon the models generated from modeling. While some specialties can begin preparations for work on an asset, none can complete the work until they receive the final model. For example, the surfacing specialty can begin work on textures for a model; however, surfacing cannot apply and adjust the textures without a model. An artist who models must understand the uses for the model by other specialties to ensure an asset will satisfy those specialties needs. The relationship is different for each specialty on the pipeline; each specialty has different requirements for an asset. However, every relationship requires communication and cooperation. Therefore, modeling must be aware of the needs of each specialty on the pipeline.

One relationship that is particularly sensitive to the construction of a real-time environment asset is the one between modeling and surfacing. Real-time rendering engines demand efficient models (clean geometry, appropriate polygon counts) to run at an acceptable frame-rate. Modeling, by itself, can produce streamlined models for the rendering engine with little to no problem; however, assets are very rarely added to a final real-time environment without first being surfaced. Hence, these models need to be streamlined by both modeling and surfacing, requiring a strong relationship between the two.
Figure 2.1 This figure illustrates three views of a production pipeline from bottom to top: A macro view of the entire pipeline, a view of one pipeline for environment art assets, and a view of the MFS process for one environment asset.
Modeling and surfacing are dependent upon one another. In the preface of *Game Development Essentials: Video Game Art*, Gantzler (2005) noted that, "[Surfacing] is often taught after modeling; however, modeling and [surfacing] are dependant on one another to some extent" (p. ix). Modeling and surfacing should not be viewed as individual processes, completed consecutively. They should be viewed as a single integrated process, completed in tandem. In the book *Game Art: Creation, Direction, and Careers*, Linde (2005) has written about the relationship between the modeling and surfacing specialties: "With the newer graphical engines, we have the ability to add details on meshes through texture render passes. Modelers should know these techniques so they adapt their model techniques to maximize the benefits of the texturing tools on their low poly models" (p. 19). Linde is advising the reader that an artist should widen their view of a model’s construction beyond the modeling specialty. Viewing the specialties outside of modeling is the premise to MFS.

MFS is a modeling method that streamlines the development of models for an environment when working on an art production pipeline. It is the combination of efficient modeling, smart surfacing, and productive integration between the two. MFS differs from traditional object modeling by allowing surfacing to affect the modeling process. Modifications to an object’s geometry, which will benefit surfacing, are considered early in the modeling process. When used appropriately, MFS can increase object production productivity by reducing errors and oversights during the modeling process.

MFS manages the overlap between modeling and surfacing through collaboration. The collaboration analyzes preset limitations from the design document, work processes, and problem areas from both perspectives of modeling and surfacing (see figure 2.1). Gantzler, while discussing a texture set applied to a piece of geometry (a wall), wrote the following in regards to collaboration: "It will be necessary to use more polygons in a wall mapped with such a set of textures, ..., but this is a worthwhile trade-off" (p. 60). Collaboration results in a give and take...
scenario. An artist analyzes the work methods and approaches from modeling and surfacing to achieve a balance between quality and efficiency.

MFS is applicable to any art production pipeline regardless of the size and composition of the development team. As long as the pipeline produces models that need to be surfaced, MFS can be exercised. MFS is a technique that can involve as few as one artist (doing both the modeling and the surfacing) or a team of modelers and surfacing artists. Observations of an art production pipeline at Tiburon showed that the modeling and surfacing responsibilities are still performed by a single position. While this may be true for Tiburon, it may not be true for all studios. However, for the sake of clarity, this thesis considers both modeling and surfacing responsibilities to be carried out by a single artist.

Figure 2.2 This figure shows the outside influences which can affect environment object modeling. It also shows which influences are involved with MFS.
CHAPTER 3

GAME DEVELOPMENT AT EA TIBURON

Research data pertaining to MFS was collected during an internship by the researcher at the Tiburon studio of Electronic Arts (EA) during the summer of 2006. The researcher used the internship opportunity to overview Tiburon's game development process and their use of art production pipelines. The pipelines were then examined for the use of MFS. Because of the researcher's limited internship duration, the research at the Tiburon studio was narrowed to observations of the Tiger Woods 08 art production pipelines. The following writing pertains directly to the Tiger Woods 08 art production pipelines and does not represent art production pipelines at other studios.

The production of a game's art assets at Tiburon are separated into five stages of development (see figure 3.1). Each stage represents a period of time (months) while also reflecting the progress of game development. A game as a whole will loosely progress from stage to stage, the progress being determined by the completion of individual assets. It is described as "loose" because individual assets can exist in stages outside of the game's current stage. For example, when a game moves from the Alpha stage to the Beta stage, it doesn't necessarily mean that all the assets within the game are ready for the Beta stage. Rather, it means that there are enough assets for the game to begin development in the Beta stage.
While a game is going through the five stages of macro-development, individual art assets within a game go through the micro-development process of production pipelines. Art assets are developed through production pipelines. A production pipeline is needed for each art asset type within a game. An asset “type” is a category of assets that can share the same production pipeline. For example, the production of an oak tree and a maple tree use the same art production pipeline because they both fall under the “tree” asset type.

**Figure 3.1**

**The Macro View of the Game Development Process**

Research, Planning, and Development → Production → Alpha → Beta → Post Production

1-3 months → 6-9 months → 1-3 months → 1-6 months

**Figure 3.2** Art assets typically move through the stages of development in a linear fashion; however, assets can be returned to previous stages at any time. The solid lines represent the typical flow of asset development and the dashed lines represent assets returned for further development.
The art asset types are determined through meetings involving producers, art directors, lead artists, and sometimes the entire team. The meetings discuss the different properties of an art asset which require the creation of an asset type. For example, a tree's complex modeling and surfacing process make it subject to the establishment of an asset type. Additionally, trees have the need to be rendered in levels of detail, which extend the tree's development process beyond that of assets like structures, grandstands, or flag pins. Once an art asset type has been established, a pipeline is created which outlines the production of assets of that type. All of this is done during the R&D phase of development.

An art pipeline at EA Tiburon, in tangible form, is a text document written by a member of an art team. Some teams dedicate pipeline writing responsibilities to one artist while others distribute the writing evenly among the team members. The document is typically a step-by-step instruction list that guides an artist through the development process of one art asset as an example for other assets of that asset type. The document may contain pictures, videos, and comments from the author. Once the document is completed, it is published to the Tiburon Intranet for internal distribution within the studio.

For example, consider the writing of a production pipeline for the "structure" asset type. The author begins by listing the parameters which outline the creation of a structure. These typically include the polygon count, texture memory limit, scale, level of detail, color palette, shaders to use, the number of UV sets, the use of the UV sets, and anything else that limit the creation of a structure. Then the author leads the reader through the creation of a structure. The author records the creation processes of modeling, surfacing, and lighting involved in the production of a structure. The recording includes detailed information needed for MFS use; photos and descriptive text explaining where the structure's geometry is modified for use by surfacing. Finally, the author lists the approximate time needed to develop an asset of the structure type, how it is placed within an environment, and where to store the files created during development.
An art pipeline, even after publication, is never considered finalized (unless mandated by someone in a management position). Art pipelines are often modified after publication. A typical scenario for this involves an artist using the pipeline to create an asset. The artist analyzes the pipeline before, during, or after asset creation, and determines that the pipeline could be modified to produce better assets. Following approval from the original author of the pipeline and sometimes a lead artist or art director, the artist will edit the document and re-post it to the intranet. A pipeline may go through this process many times, most often during the R&D stage of development. However, a pipeline may be edited during later stages.

Research, concept development, and planning

Every game developed at EA Tiburon begins with the research, concept development, and planning stage (R&D). The R&D stage is for creating working plans for the game, testing new ideas to go in-game, creating production pipelines, testing the production pipelines, learning to use new technology (if necessary), and problem solving. Most of the work generated during R&D will be used by the other phases, making R&D the most important stage of development. A game will spend somewhere between one to three months in R&D.

In the case of Tiger Woods 08 the R&D phase played a huge role for the development team. The Tiger Woods franchise had been produced at EA's Redwood Shores studio in 2006 and 2007, and was moved to Tiburon for production beginning with the 08 game. EA expected the Tiburon studio to take the previous work completed on Tiger Woods 06 and Tiger Woods 07 and continue the franchise's growth. Because of the game's transfer between studios, the Tiger team was granted a much longer R&D development stage. Extra time had to be budgeted for bringing individuals onto the team, getting those individuals organized and prepped for work, and for exposing the newly created Tiger team to the production pipelines used by the Tiger team from Redwood Shores.
The Tiburon Tiger team began work by analyzing the 06 and 07 Tiger Woods art assets. This helped the Tiburon Tiger team to understand the scope of the Tiger Woods franchise. Decisions could then be made for changes to production pipelines, changes to the game (from art assets and programming to game play and presentation), and improvements or new additions to the game. This is the point where the Tiger environment art pod became routinely involved with R&D for the game.

The environment art pod recognized that time for R&D was limited, even with the extended R&D phase, so decisions had to be made about prioritizing art asset pipeline development. The largest factor to effect the R&D phase was the technology transition from the 2000-2001 video game consoles to the 2005-2006 video game consoles. The new consoles offered opportunities for art asset development that many of the artists had yet to explore. The new consoles improved visuals through advancements in hardware technology. Most importantly were the ability to use programmable shaders and advanced lighting techniques. The combination of this with a new unfamiliar game pushed the importance of the R&D stage for the environment art pod. Thus, the Tiger environment art pod’s main task during R&D was to analyze the environment art production pipelines used in Tiger Woods 06 and Tiger Woods 07 and adapt them or rebuild them for use in Tiger Woods 08.

The Tiger environment art pod was also testing new modeling and surfacing approaches for the many objects within the game. During these tests the environment artists began to look at how MFS could streamline an object’s development. For example, the Tiburon Tiger team was not satisfied with the look or the production tools and pipelines used in developing the trees in the 06 and 07 games. Artists, including the researcher, were given the task to research new production methods that would improve the realism of trees and streamline their development.

The solution to the tree problem had to meet several criteria. First, the solution had to improve the visual quality of the trees without taxing the hardware. The art director wanted the trees to
look more realistic in their modeling, surfacing, and shadowing but did not want the trees to use too much of the GPU’s processing power. Finding this solution involved the use of MFS to create efficient geometry that worked with the shaders written for the tree’s surfacing (see chapter five for shaders). Second, the solution had to allow for levels of detail while maintaining a high level of visual quality. Third, the solution had to allow the artist the ability to create unique trees to match those seen at real-world golf courses. Fourth, the solution had to allow the artist to create multiple trees of different types, ages, and sizes in a time efficient manner. Finally, the solution had to work with the physics and collision detection programming to be used in the game.

Production

While the R&D stage sets the foundation for a game’s development, the production stage builds the framework. Production is typically the longest stage because the majority of the game’s art assets are created during this time. A game developed at Tiburon will spend six to eight months in production.

Asset development during the production stage is very habitual. Environment artists use design documents to ensure the development of comparable assets, which becomes a routine process. Including MFS data in the design document helps an artist to efficiently develop all assets using that document. This becomes important when the assets are assembled and analyzed during the alpha stage. When MFS data is not included with the design document, it is the responsibility of the artist to use MFS in the asset’s development. Using MFS is related to an environment artist’s experience; those with more experience tend to use MFS more often.

All work during the production stage brings the game closer to completion. Implementing new ideas (known as feature creep) are avoided because they can require lengthy backtracking or, in the worst case, a complete restart of production. Anything that distracts from the production
of the game’s art assets is postponed to the alpha stage. By the end of production, the game’s art assets are ready for evaluation in the game engine.

Alpha

The alpha stage combines the art assets created during the production stage for evaluation in the target platform's game engine. Alpha is also an opportunity for producers and art directors to see how the art assets are working with the game features, game play, and presentation. During alpha, art assets are often sent back to production for small modifications like scale inconsistencies. In some cases, art assets are returned to R&D. Returning an art asset to R&D usually means that the asset did not function correctly in the game, it was too heavy a burden on the frame rate, or its function in the game was changed. A game at Tiburon will take one to two months to progress through the alpha stage.

For art assets, alpha grants artists an opportunity to view the assets (models and textures) running together as a whole in the real-time environment. This is where the MFS process shows its value. Artists check the performance of assets against the target platform’s frame rate (the accepted average frame rate for all real-time games and applications is at or above 30 frames per second). Artists then evaluate changes to models or textures that will improve the frame rate (if the frame rate is too low) or enhance the visuals in the scene (if the frame rate is well above 30 fps). By using MFS, the artists give themselves the best chance to avoid frame rate problems.

Alpha is also a time when the game is tested for errors, bugs, and oversights. Testing a game is a lengthy process that involves the quality assurance (QA) team. They report any and all problems with a game back to the development team. Problems typically range from programming glitches to game-play inconsistencies. However, the art director usually catches problems with the art before the game is handled by QA. After the problems caught by QA are fixed and the game’s development is at its first playable state, the game will move to the beta stage of development.
Beta

Once the problems discovered during the alpha phase have been addressed, the game progresses to the beta stage. When a game is labeled as "beta" it means the game is at a near finished, playable state. One could legitimately take the beta version of a game and play it without errors. The beta stage is implemented to check that art assets are working correctly, that the game is running smoothly, and that the presentation matches the producer's and art director's ambition. A game at Tiburon will spend about a month in the beta stage.

During beta the environment artists are putting the final touches on assets like cleaning up textures, checking the placement and inclusion of objects within the environment, and watching for color and lighting uniformity. MFS has little involvement in the beta stage because most of the art asset development has moved beyond production. However, MFS can be used to remodel or resurface an asset that is a frame rate burden.

At the end of beta, and after approval from the producer and art director, a game becomes "gold." The game is now considered complete and ready for mass production. Even though the game is completed, development on the game does not always stop.

Post Production

The post production stage is for any art assets developed after a game has been released. In recent years, post production has evolved into an extended production stage. Game developers will purposely create content during post production that is intended for post release consumption. The content can range from updates to fix problems missed during testing to completely new assets like environments and characters. The art assets created during post production go through the first four stages of development, except instead of going gold they are packaged into an update. The update is then made available for download through the Internet.
A game is not required to have a post-production stage. The decision happens per game and is made by the executive producer. Typically, a game will go into post production at Tiburon if the producer believes the game can make additional profit from the sale or use of the post production content. With games that require a subscription fee, or use in-game advertising, post production content is used to entice the consumer into continuing their play with a game.
CHAPTER 4

THE ENVIRONMENT ARTIST

At EA Tiburon, the tasks of creating objects and environments are performed by an individual known as the "environment artist". The environment artist holds a unique position because of the wide range of required skills. They can be responsible for the modeling, surfacing, light baking, and occasionally rigging and animation of an object (They do not work on characters, that is another team). Tiburon chooses to hire environment artists rather than modeling and surfacing specialists. Ron Amador, the lead environment artist from Tiger Woods 08, said the following about employing an environment artist versus individual modelers and surfacing artists: "[Tiburon] eliminate[s] potential issues [problems between modeling and surfacing] by having modelers that are also skilled surfacers."¹

An environment artist at Tiburon spends the majority of their time modeling, surfacing, and light baking objects for environments. For Tiger Woods 08, the environment artists work to create the golf courses used for play within the game. Each course includes 18 playable holes, houses, buildings, and clubhouses, vegetation like trees and bushes, and scenic objects like deserts, oceans, lakes, fields, or swamps, and many more objects that are too numerous to list. The successful completion of environment objects is a daunting task, challenging but not impossible. The environment artists overcome the challenges through organization, teamwork, and communication.
The environment artists are organized into a space known as a pod. A pod is simply a physical grouping of cubicles arranged to encourage the flow of communication and cooperation. At any time artists within the pod may collaborate to solve a problem. This happens quite often, usually in groups of 2-3. Gathering fellow environment artists is beneficial because it creates an arena for critiquing ideas. These “mini meetings” improve productivity by allowing the artists to address problems as they arise rather than having to schedule and wait for a formal meeting. They also provide an opportunity for the artists to share their individual failures and successes with MFS.

Each artist uses MFS differently because MFS is a technique learned through experience. By collaborating MFS knowledge and experience, the artists learn to work more efficiently. One environment artist, when questioned about the helpfulness of collaboration, said the following: “Maya/3D packages have numerous permutations in the process of achieving a specific product. Different approaches may result in the streamlining of a process which can improve efficiency and overall end product.”

In addition to teamwork between environment artists is the teamwork between environment artists and technical artists (see chapter 5 for details about the technical artist). The two share a special relationship which exists because of the complexities in technology that are part of modern video game development. The technical artists use their knowledge of the technology to support the environment artists. This involves just about every aspect of work within the pod; from file management and resource organization to software scripting and programming. The most significant support from the TA comes in the form of shader writing. Shader writing plays an important role in the planning and creation of models, which in turn has an impact on the MFS process. The impact of shader writing on the MFS process is discussed in detail in chapter 5.
Managing the successful creation of environment objects is a task left to the lead artist. The lead artist is typically the environment artist with the most experience or with the most outstanding organizational and leadership skills. He delegates art assets for creation throughout the environment pod, calling on the strengths of individuals for different assignments. The lead artist also has production tasks, usually along the lines of production pipeline research.

The single most important responsibility for the lead artist is to ensure the creation of art assets through art production pipelines. It is the responsibility of the lead artist to recognize art asset types that require individual pipelines and to assign the pipeline creation to either himself or another environment artist within the pod. The production pipelines allow his environment pod to work quickly and efficiently through the thousands of art assets created during the production phase. The number of pipelines for each environment is dependent on the nature of the game. Sports games, like Tiger Woods 08, require fewer pipelines because they require only one environment type (a golf course for Tiger Woods 08, a football stadium for NCAA Football 07, etc.). The golf course environments in Tiger Woods 08 are similar enough so that pipelines can be shared across the courses.

In the case of Tiger Woods 08, the lead artist identified a need for four different object creation pipelines. One for the terrain, a second for the structures on a course, a third for trees and bushes, and a fourth for tournament objects such as observation towers and grandstands. These categories were needed because the pipelines could not be shared among the four. While artists were designing the pipelines, the lead artist commissioned another artist to research surfacing and shader application and another to research light map techniques. This was done to understand how far the hardware technology of the target platforms would allow improvements to surfacing and lighting.

Communication is key to the organization and teamwork within the pod. The Tiger Woods 08 environment art pod has its own website within the internal Tiburon Intranet just for this pur-
pose. It is a portal to quickly and widely share information with other artists in the pod. The website performs multiple functions for the pod. It is used to store and supply the production pipeline, file management, task management, and other documents used by artists within the pod. The site is also viewed by art directors and producers; this gives them an opportunity to stay informed of the progress of the pod without scheduling lengthy progress report meetings.

The Tiger Woods 08 environment pod also maintains a storyboard wall for an at-a-glance overview of the progress being made on an environment. This helped the environment artists to judge the speed of their work, and to adjust accordingly.

END NOTES

1. See the third survey in appendix C, page 103
2. See the first survey in appendix C, page 89
CHAPTER 5

THE TECHNICAL ARTIST

During the internship at Tiburon, the researcher observed the existence of an additional member of the environment art pod who had a direct impact on the MFS process. The additional member was the relatively new position of technical artist (TA). This position was added during the video game console transition from the 2000-2001 generation to the 2005-2006 generation.

The position of TA evolved from the constantly changing world of real-time game development. Changes in technology, tools, and skill sets necessitated the creation of the TA position. The TA’s purpose is to mediate the complexities of technology and tools for artists. They are the utility men of the pod. If something breaks, they fix it. When a problem arises, they often find the solution. The TA’s responsibilities span from the technical side of programming to the creative side of shader writing.

The TA’s most direct impact on the MFS process comes from shader writing. A TA works with an artist to write a shader, usually to define the surfacing of an asset. The artist communicates a list of needs for the shader to the TA. The TA writes the shader based on the artist’s specifications, checking that the shader is making efficient use of the target platform’s GPU. When the TA is satisfied with the shader, the shader is given back to the artist for evaluation with an asset
in a real-time environment. After evaluation, the TA and artist collaborate to fine-tune the shader.

For example, an environment artist meets with a TA to discuss the creation of a new shader for sand traps. At the meeting, the environment artist describes the desired functionality from the shader asking for texture map inputs, procedural texture controls, and other UI elements such as UV map controls. For the sand trap shader, the environment artist asks for two color map texture file inputs and one normal map texture file input. He asks for controls for the tiling properties of all the texture maps. He asks for tint controls for both the color maps. He asks for a depth control for the normal map. He asks for controls over the shader’s procedural noise, including controls for size, density, intensity, and the arrangement of noise. Finally, he asks for the ability to use two UV sets. From this list the TA writes the shader.

Once the shader is written it is given to the environment artist for evaluation. The environment artist checks the sand trap shader’s performance within his working environment. He links in proper color and normal maps, adjusts UI controls, and assigns UV sets within the shader’s UI controls. During the evaluation process, the environment artist may ask for changes to the shader’s UI or to its functionality. If this occurs, the environment artist and TA will often work side-by-side to calibrate the shader for the desired performance.

For successful asset production, TAs and artists communicate and collaborate in a similar manner to that seen in the MFS process. In this relationship, the TA’s role is similar to modeling and the artist’s role is similar to surfacing, in the MFS process. The shader, like a model, has a large impact on the surfacing approach for an asset. Because of this, artists meet with TAs to discuss the properties of a shader before it is written. This is analogous to an artist considering the modeling and surfacing needs of an asset before modeling begins.
The discovery of the TA position was unexpected, but important. The impact shaders have on modeling and surfacing is substantial, as seen in the survey response from an environment artist.

I will often collaborate with a TA to plan and create a shader that will be able to work with the object that I am modeling. I will often use the knowledge of how a particular shader works in order to plan on how I will model, layout UVs, and texture an object. So much of my plan on the creation of a complicated model depends on the way I plan to use a particular shader. I have often reworked a model ... if a shader is updated or has to change... ¹

The environment artists are just as dependent upon shaders as surfacing is dependent upon modeling in the MFS process. This observation was not expected by the researcher and came at a great surprise. The demanding nature of real-time rendering places such a burden on efficient asset development that everyone involved with asset development must be efficient minded. If Tiburon were to use modelers, surfacing artists, and TAs the chances for poor asset development would increase because of the difficulties in managing efficiency across three individuals with separate tasks. Also the time lapses between art asset handoffs between the three would most likely slow down production.

END NOTES

1. See the second survey in appendix C, page 96
CHAPTER 6

SURVEY ANALYSIS

The environment artists from the Tiger Woods 08 team were surveyed for their knowledge and use of MFS on present and past games. Five environment artists completed a written survey, including the lead artist. The responses from the surveys can be viewed in appendix C. The questions cover five areas related to MFS:

1. The modeling to surfacing pipeline
2. Collaboration
3. Design documents
4. UV layout
5. Miscellaneous

Questions that were left unanswered, were not relevant to MFS (like questions about the environment artist's name), or had answers of 'N/A' are not included in the analysis.
The modeling to surfacing pipeline

Question 3a. Briefly describe the flow of content within your current art production pipeline. (ex. modeling > surfacing > lighting > etc.)

The responses were nearly identical throughout the sample. All participants placed modeling ahead of surfacing and surfacing ahead of lighting. One interesting difference between the responses was that between the senior and junior artists. The junior artists described the pipeline linearly where as the senior artists described some areas of the pipeline as cyclical or parallel.

Question 3b. Describe the modeling to surfacing process within your current art pipeline.

Once again, there was a distinct divide between the junior and senior artists. The junior artists again answered the question by describing the MFS process as linear; the senior artists described the MFS process as “concurrent development.”

According to the artists, modeling comes before surfacing and lighting on the art production pipeline. Experienced artists often integrate surfacing into the modeling process to allow for concurrent development. The divide between the senior and junior level artists from question 3a and 3b are likely due to differences in experience. The senior artists have modeled and surfaced more assets over their career, giving them more opportunities to see how surfacing can influence modeling.

Collaboration

The majority of the questions pertaining to collaboration were left unanswered. This was due to the fact that the survey was designed for modelers who work with surfacing artists and not environment artists who do both. However, there were some notable results.
Question 6. Do you find collaboration helpful in the modeling process?

Four artists responded, three who found collaboration helpful and one who did not. The responses were tailored towards collaboration between environment artists and not between modelers and surfacing artists. Two of the positive responses said that collaboration generates ideas and good feedback and streamlines processes. Interestingly, the artist who responded negatively towards collaboration said this was because "many modelers have different approaches and levels of competency." ¹

Question 7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

never            sometimes          often           always

The lead artist responded with “never,” two artists responded with “sometimes,” and one artist responded with “often.”

The data collected from these questions is skewed because the Tiger Woods 08 art production pipeline uses environment artists. None of the artists answered the questions as a modeler working with a surfacing artist. However, the environment artists still collaborate in asset development, with the majority finding collaboration helpful. Only the lead artist said that surfacing needs never caused the remodeling of assets. The others said that surfacing needs can at least sometimes cause an asset to be remodeled.

Design document

Question 10. Do you receive a design document before you begin modeling?

All but one artist responded with “yes.” The artist who responded “no” said that he typically did not receive design documents because it was his responsibility to create them.
Question 11. How does the design document aid you with modeling?

The artists found the design document provided a baseline or guidelines for the development process. This included the specifications for the asset (texture memory, polygon count), the production pipeline instructions, and advice for successful asset production.

Question 12. Does the design document aid you with determining which details to model and which details to pass to the surfacing artist?

Yes

If no, how are the details resolved?

The response was a varied "no." The artists resolved the modeling or surfacing problem by pursuing one of the following options: the artist's interpretation and personal experience, discovering the solution through R&D, or discussing the issue with the art director.

Question 13. Have you ever strayed from the design document to design a better object?

If yes, how did that effect the object's movement down the production pipeline?

All but one admitted to abandoning parts or the entire design document during art asset production. One said, "The current work environment encourages design [document] changes that improve workflow. Improvement in speed and work quality allowed time to be spent on other areas that needed attention." Another said, "Most of the time, if I plan to stray from the design [document] and feel it may have possible impact down the pipeline, I will check with [technical artists], programmers, etc. to determine possible drawbacks."

Most of the time, the environment artists create the design documents for environment art on Tiger Woods 08. They receive pre-visualization images from the art directors, which give them a visual target for the game. Then an asset type is established (ex., structures). An artist(s) researches the production of that asset type, documenting the production process (the pipeline) and the specifications for that asset (texture memory usage and polygon count). The design
document is by no means the required approach for the production of an asset. Often times the environment artists stray from the design document to design a better or more efficient model.

**UV layout**

The survey had two lines of questioning regarding UV layout. One line posed questions to a modeler who was responsible for the UV layout of a model. The second line posed questions to a modeler who was not responsible for the UV layout of a model. Because the sample group consisted of all environment artists, they all followed the first line of questioning.

Question 16a.) At what point in the modeling process of an organic object is the UV layout considered?

Three artists reported the UV layout was considered after modeling was completed. One respondent said that UV layout happened concurrently with modeling. Another respondent said that UV layout was considered throughout the modeling process.

Question 16b.) Is the UV layout of an organic object considered more complex than the UV layout of a planar object?

When laying out UVs, all of the artists found organic objects to be more difficult than planar. This was due to the nature of the organic object and its larger potential for seaming and texture flow issues.

Question 16c.) Can the complexity of the UV layout of an organic object effect/change the modeling approach for that object?

Yes  No

Three out of five respondents agreed that the complexity of the UV layout of an organic object could affect or change the modeling approach.
Question 16d. Do you consider the UV layout of an object before modeling?

If yes, how?

If no, why?

When asked about the consideration of UV layout before modeling, a conflicting four out of five responded positively. Two of the artists who reported in question 16a to post modeling UV layout consideration said they planned or thought about the UV layout before modeling.

Question 16e. Are you in the practice of laying out the UVs of an object during the modeling process?

If yes, why?

If no, why?

Only two out of five responded positively. The negative respondents, all junior artists, found that UV layout during object modeling was inefficient because of frequent changes to the model's geometry. The layout of UVs can be a lengthy process and doing it after every modeling change can be tiresome as well as inefficient. The two positive respondents came from senior artists. They both found UV layout during modeling beneficial when that object has repeating or iterative features. Doing so improved the speed and aesthetic output of the objects.

Question 16f. Can the layout of the UV's of an object after modeling be difficult or time consuming?

Yes No

Two responded with “no,” two with “yes,” and one with “sometimes.”

Question 18. What are the general problems with the UV layout when a remodel is requested?

All of the artists reported that changes in the model's shape, especially around border edges and seams, require re-UV layout. One artist said that doing so could add significant time to object production.
UV layout is considered before, during, and after modeling, depending on the experience of the artist. Organic objects are more difficult to UV than planar. Most of the artists are aware of the UV concerns while they model; however, not all of the artists adapt their modeling for those concerns. There is a divide among artists about the difficulty of UV layout after object modeling, most likely due to differences in UV layout experience. The biggest problem with UV layout is the constant change to a model's geometry. Those changes often postpone the UV layout to post modeling. The layout of UVs during modeling is found to be good for repeating details such as windows or columns.

Miscellaneous questions

The final set of questions asked participants for their opinions about preparations for work on an art production pipeline. It also gave them an opportunity to share any other information they thought to be significant.

Question 19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

Yes  No

If Yes, why?

All of the respondents replied with “no.” Two artists said that personal experience plays a role in adapting to new art production pipelines. One artist said that, “...experienced artists can foresee issues at every step of the process and learn to avoid them.”

Question 20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

Yes  No

Four out of five responded with “yes.”
Question 21. In your opinion, what surfacing knowledge must a modeler possess to be able to model objects for an art production pipeline?

The responses to the question spanned the surfacing continuum, requiring modelers to have a surfacing knowledge somewhere between a little to everything. One artist answered that a modeler only needs to understand the constraints surfacing can place on a model, where as another artist answered that modelers need to know everything about the surfacing process. The latter artist also said that knowing the entire surfacing process is a huge help when surfacing and modeling are handled by individual artists.

Question 22. In your opinion, what modeling knowledge must a surfacing artist possess to be able to surface objects for an art production pipeline?

The responses for modeling knowledge needed by surfacing artists were divided. Half of the sample said surfacing artists needed "little to none" in terms of modeling knowledge. They found that a surfacing artist should only need to understand how the object's geometry is used within the 3D space. They also believed a surfacing artist's skills fall more in line with a traditional visual artist (illustration and painting). The other half of the responses found that surfacing artists do need modeling knowledge. One artist said that surfacing artists should be able to optimize the geometry of an object for surfacing. Another respondent believed surfacing artists need at least an awareness of the processes, tools, and procedures involved with an object's construction.

Question 23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.

Two of the artists saw the individual roles of modeler and surfacing artist disappearing or merging into a single position. The reasons for this merge were changes in technology requiring different skill sets from artists and the formation of completely new positions. One artist summed up his MFS experience as follows: "A concurrent and iterative approach to modeling
and surfacing has produced the best results from a personal perspective — It allows for a progressive review and reduced amount of rework that needs to be done on the final product.  

The responses from question 19 are surprising. The researcher had expected the artists to agree that new hires or interns would have difficulty moving assets down the art production pipeline. The problem existing from the inexperience of a new artist and the complexities of art production pipelines.

The responses from question 19 also conflict with the responses from question 20. How can the artists believe that new hires and interns would not have difficulty moving assets down the art production pipeline but also believe that a company would benefit from hiring new artists with a working knowledge of art production pipelines?

The conflicting response to question 19 may have resulted because of two factors. First, the environment artists may have answered the question by reflecting upon the researcher’s own experience with adapting to the art production pipeline at Tiburon. The researcher had little difficulty adapting to the art production pipelines because of his research work prior to the internship.

Second, the artists had long been participating on art production pipelines. They may have forgotten about any difficulties with creating assets on their first art production pipeline. Their first pipeline could also have been considerably less complex due to the hardware technology of the time (all of the artists had at least 5 years of experience).
END NOTES

1. See appendix C, survey five, page 117
2. See appendix C, survey one, page 89
3. See appendix C, survey two, page 96
4. See appendix C, survey three, page 103
5. See appendix C, survey one, page 89
CHAPTER 7

MFS CONCEPTS AND SCENARIOS

Determining when, where, or how to use the MFS process can be difficult for the inexperienced artist. This is because the problem solving process for determining the correct MFS approach is not always clear or obvious. There are two options to ameliorate the problem solving process. First, an artist can draw from a set of MFS concepts. These concepts guide an artist towards a possible MFS approach. Second, the artist can draw from past experiences; a scenario exists from each successful use of MFS by the artist, or from other artists with which one may communicate. By analyzing MFS concepts and looking through past scenarios, an artist can overcome their inexperience and be confident in finding a solution during problem solving.

Understanding the concepts and scenarios begins by understanding the connections between modeling, UVing, and surfacing within MFS. Modeling, UVing, and surfacing all share a reciprocal bond so that each is dependent on or influenced by the other to some extent. An object must be modeled before it can be UVed and the object must be UVed before it can be surfaced. So surfacing is dependent upon UVing because the UV map dictates the appearance of textures on an object. UVing is dependent upon modeling because UVs cannot exist without geometry to define them.

While modeling appears to be the dominant component, it is actually the most submissive within MFS. This is because of the influence surfacing holds over UVing and the influence
UVing holds over an object's geometry. Surfacing holds heavy influence over the UV layout because the UVs determine how a texture appears on an object: the UVs must be arranged to display a texture correctly. UVing holds a heavy influence over modeling because an object's geometry, specifically vertices and edges, define the existence and arrangement of UVs. An object's geometry must be modified to add or remove UVs. Thus, the flow of influences is in the same direction as the flow of dependencies (see figure 7.1).

The bonds between modeling, UVing, and surfacing can also be seen when modifications are made to either of the components (modeling, UVing, and surfacing). Changes made to an object's geometry will affect the UV map, which will affect the appearance of the surfacing. If an artist removes an edge loop (vertices included) from an object's internal geometry, the UVs that represent that edge loop are also removed. This can cause the appearance of the surfacing around that edge loop to change (see figure 7.2).

![MFS Component Connections](image)

Figure 7.1 The flow of dependencies and influences between the three components of MFS.
The texture's appearance in relation to the cube's geometry

The cube's UV placement in relation to the texture

Figure 7.2 Edges one and four in the bottom half of figure 7.2 are exaggerated to show their location. They actually share the same location as edges five and six.

Concepts

Now that the dependencies and influences between the components of MFS have been expounded, the concepts behind MFS decision making can be illuminated. Illuminating the decision making process starts by understanding several key MFS concepts. The concepts take the form of questions asked of an artist while determining the correct MFS method.

What limitations exist that restrict the development of the object?
The first concept is less of a question and more of an information gathering quest. An artist must discover the parameters (usually described in a design document) that govern an asset's development. This includes, but is not limited to, the polygon limit, the texture memory available, how many UV sets can be used, shader requirements, and the object's visual importance.
in the scene. These requirements are unavoidable and typically unchangeable, so an artist must adapt the MFS approach around them.

An artist uses the limitations to narrow the MFS approach possibilities, or eliminate the use of MFS altogether. The first limitations to examine are the polygon and texture memory restrictions and the object's importance in the scene. These usually go hand-in-hand. If an object has a low visual priority in the scene, it receives a smaller polygon and texture memory budget and vice-versa. Objects with low visual priority or low polygon and texture memory usage are often exempt to MFS because of their simplicity. However, the use of MFS cannot be ruled out until the shader and UV set restrictions are examined.

The next step is to look at the shader and UV set restrictions. This is typically where an artist can determine if MFS is needed. If the object uses a simple shader and requires only one UV set (e.g. a shader that maps only one color texture to one UV set), there may be no need to use MFS. However, when an object requires multiple UV sets or has a demanding shader, this is a good indication that MFS should be used.

Once an artist has an understanding of the initial limitations, additional questions are asked to further narrow the MFS approach. It is best to start with surfacing questions followed by UV and then modeling questions because of the flow of influences.

**What method of texture painting should be used?**

The answer to this question may be simple, but it has a huge impact on the modeling and UV-ing methods. There are two likely answers: use a 2D or 3D painting software package. Both are perfectly suitable for creating textures but each have their strengths and weaknesses.

2D painting software is great for texturing planar objects. The angular nature of planar objects fits well with textures painted in 2D. This is because seams between texture edges are
less noticeable and easier to manage. 2D is also the better method for creating tileable textures because of the tools included with most 2D paint programs.

Using 2D painting software has its disadvantages as well, with two major drawbacks. The first is a heavy dependency on UV layout. Because the textures are painted in a separate software package separate from the modeling package, the textures must be applied through managed UV layout. The UV layout process can be cumbersome and time consuming. Second, the artist experiences the discontinuity of moving between the 2D painting software and 3D asset creation software. This minor inconvenience becomes a major annoyance when going back and forth between two programs for the tuning of a texture.

3D painting software is great for texturing organic objects. 3D painting software packages allow the artist to paint directly on the object's surface, and avoid the process of UV layout management. The UVs are arranged into their zero to one space in a fashion that avoids overlap (this can easily be done by choosing the automatic mapping function within the 3D asset or 3D painting software). This in turn allows an artist to paint directly on the surface of an object without running into seaming problems. However, there are drawbacks to using 3D painting software as well. Because the UVs are automatically mapped to the zero to one space, only textures created from the 3D painting software can be used to surface that object. This also removes the option to use tileable textures. 3D paint programs also have a tendency to create larger file sizes because they need more pixel space to convey details.

The impact of using 2D and 3D texture painting on the MFS approach is significant. If the 3D painting method is chosen, MFS has little impact on the creation of the asset. The object can be modeled, UVed, and surfaced with little concern for the flow of influences because it does not matter. The artist can use any modeling method to generate the geometry for the object; the artist does not have to be concerned about the object's subdivision for a proper UV layout, and the UVs do not have to be managed to correctly display the textures. If the 2D method is
chosen, MFS has a large impact on the creation of the asset. With 2D texture painting, textures are painted and then applied to a model. To apply the textures the UVs must be managed, and managing the UVs for proper texture display can require modifications an object's geometry. This, of course, is following the flow of influences which is central to MFS.

Choosing the texture painting method can be dependent upon the shape of the object or its method of UV layout, which leads to the next question.

**Does the object's shape facilitate or hinder the UV layout process?**

This question is important for determining the point at which the layout of UVs happens. The layout of UVs for an object can happen during modeling or after; the choice is the preference of the artist. However, there are objects that are incredibly difficult to UV after modeling but easy to UV during modeling. These objects are identified by their inability to be UVed after modeling through traditional mapping methods like planar, cylindrical, or spherical. If the artist is aware of the impending UVing difficulties post-modeling, he can take care to look for opportunities to UV during modeling.

UVing during modeling is just that: modeling and UVing in tandem. An artist models, then UVs the modeled area, then models again, then UVs the modeled area, and so on until the object is complete. It does not have to be a 1:1 process; the artist needs to keep a watchful eye for modeling actions that can result in difficult UVing.

For example, imagine an artist is modeling a pipe and plans to use a tileable texture to surface the pipe. The artist wants the pipe to have two 90 degree bends. Should the artist layout the UVs of the pipe before or after bending its geometry? In this example, the artist should UV map the pipe before bending. This allows the artist to use the cylindrical UV projection function which is quick and easy. When the bends are made, the tileable texture continues along the pipe because the bending of the geometry does not affect the UVs (only adding or removing
vertices, edges, or faces can change the UV map without directly modifying the UVs). If the artist waits to UV after bending the pipe, he has to use three cylindrical UV projections (one for each bend in the pipe) and then sew the UVs of the three projections together. Performing the two extra UV projections in post-modeling does not add a significant amount of time to the UVing process, so does it really matter? Maybe not for two extra bends, but imagine how long it would take to make 49 extra projections for a pipe that bends 50 times, or to model a set of 50 different pipes with three bends each. The point is that an artist can save time by UVing during modeling.

**Which type of geometry best suits the desired surfacing approach?**

Choosing to model with polygons or NURBS is based on the type of textures an artist expects to use when surfacing an object. Tileable textures work well with NURBS because the NURBS’ texture space is defined to be within the full zero to one range. This ensures the continuous flow of the tileable texture across the NURBS geometry (after separating a NURBS geometry into patches). Non-tileable textures work well with polygons because NURBS have a tendency to stretch textures (once again, a NURBS’ texture space is defined to fit within the full zero to one range).

NURBS geometry is unique in that it uses curves to define the geometry’s surface shape instead of vertices and edges. NURBS geometry is also unique because it does not have UVs to position and display textures. With NURBS, the texture space is always set in the full zero to one range and the texture space cannot be directly manipulated. The only way to change the texture space for a NURBS object is to separate the geometry into patches. Once separated, each patch automatically has its texture space redefined to fit within the full zero to one range.

Possibly the biggest advantage to modeling with NURBS is the time saved in texture application. With NURBS, an artist has no control over the texture coordinates within the texture space; in other words, there is no UVing. Also, if the artist needs the control that UVs grant,
the NURBS geometry can be converted to polygons while preserving the appearance of any textures already applied.

So far, four major questions have been posed to reveal surfacing-, UVing-, and modeling-related MFS concepts. There are also several minor questions with answers that supplement the concepts already discussed.

**Is an object surfaced with textures that can be used by multiple objects?**

This question is dealing with the concept of texture management. Texture management is important for real-time because the target rendering platform always has a texture memory limit. Once the texture memory limit is reached, no more textures can be loaded into a scene without unloading others. Because of this limitation, it is common practice in real-time to create commonly used textures that can be shared among several objects.

MFS supports proper texture management because it is a process of efficiency. When using MFS, an object is modeled and surfaced in such a way that its geometry supports commonly used textures. It also helps to prevent the need for one-use textures (a texture used only by one object). An example of texture management is presented in chapter eight.

**Can the geometry be separated into smaller pieces?**

Separating geometry, not to be confused with subdivision, is a good way to improve modeling efficiency. When modeling for real-time, it is important to create clean geometry free of t-junctions, n-sided polygons, and stray vertices (see appendix A). To fix these modeling problems without using separation, an artist must add or adjust the geometry in such a way that reduces efficiency.

Separating the geometry is especially helpful when dealing with t-junctions. T-junctions are a problem when they appear within an object's internal geometry. Fixing a t-junction usually
Resolving T-Junction Problems

Separated geometry solution

Non-separated geometry solution

Figure 7.3 The spaces between the pieces of geometry in the separated geometry solution are exaggerated to convey the idea of being separate. In actuality, the edges of the separate pieces of geometry abut one another.

involves connecting the vertex at the center of the t-junction with another vertex adjacent to it. However, t-junctions can exist on the borders of geometry. If the geometry is separated along the edge that contains the t-junction, no extra geometry is needed to fix the t-junction (see figure 7.3).

Which details of an object are modeled and which details are conveyed through surfacing? Choosing which details to model and which details to surface is influenced by many factors, including the design document, the object’s visual importance, and the polygon and texture memory limits. These factors complicate the answer to the question, but there are some basic concepts that can aid the artist.

Each detail of an object can be expressed through modeling, surfacing, or a combination of the two. The correct approach depends on the nature of the detail. For example, if an artist is creating a crate with chicken wire walls and a wooden frame, the artist has three options to create
the crate. First, he can model both the wooden frame and the chicken wire and then surface each with a tileable texture. Second, he can model and surface the wooden frame similarly to the first option, then apply a chicken wire texture to a single quad for each wall of the crate. Third, he can model the crate using a simple primitive cube, then apply a texture containing both the wooden frame and chicken wire details to each side of the cube.

The optimal approach for this example is option number two. Option one wastes polygons by unnecessarily modeling the chicken wire. Option three is a possibility, however, the efficiency gains in using a primitive cube do not outweigh the reduction in visual quality. Also, because the wood and chicken wire details are combined into a single texture, the texture becomes a one-use texture and can only be used for other chicken wire crates. Option two blends the best use of modeling and surfacing to express the details about the chicken wire crate.

Scenarios

For this thesis, a selection of MFS concepts are examined through scenarios to give a practical understanding of their use. Each scenario presents an object for modeling and surfacing and demonstrates how to complete the object through some means of MFS.

Scenario one: the garden hose

The objective for this scenario is to model, UV, and surface a garden hose in a timely fashion. Two modeling approaches are examined, modeling with polygons and modeling with NURBS, so that a comparison can be made between the advantages and disadvantages of the two. Both modeling methods use an extrusion modeling method, so both are starting with the same curve to extrude along (see figure 7.4). Both methods also use a tileable red arrow texture for the surfacing.
Figure 7.4 The curve for the garden hose extrusion.

Figure 7.5 A polygonal cylinder is used as the base for the polygonal garden hose extrusion.
Figure 7.6 The cylinder is extruded along the curve to create the polygonal geometry for the garden hose, then surfaced with a red arrow texture.

Polygonal extrusion requires a curve to extrude along and a polygon face to define the shape of the extrusion. The curve has already been set and a primitive cylinder is appropriately chosen as the extrusion shape (see figure 7.5). With help from the extrusion function of the modeling software, the geometry of the hose is quickly created. After the extrusion, a simple lambert shader with an arrow color map is assigned to the geometry (see figure 7.6). The red arrow texture is used because it clearly indicates the organization of the texture space coordinates in relation to the actual appearance of the texture on the geometry.

At this point the first problem with the polygon extrusion modeling method becomes apparent. The red arrow texture is stretching along the length of the geometry rather than repeating. The cause is a problem with the UV layout that results from the polygonal extrusion. The polygonal extrusion function does not automatically place UVs in a usable manner.
A look at the texture space coordinates reveals an unwelcome sight (see the right half of figure 7.7). The railroad pattern of UVs are those from the original cylinder used for extrusion. The remaining UVs of the polygonal garden hose are difficult to see because they share the same texture space location as the top row of UVs from the cylinder. Nearly all of the UVs are stacked atop one another on the top row of the railroad pattern instead of spaced apart like the original cylinder.

The UVs are unmanageable in this state. Adjusting them into a tileable state is a lengthy and monotonous task that cannot be alleviated by one of the UV mapping functions. Neither the planar, cylindrical, spherical, or automatic UV mapping functions can quickly resolve the problem because the garden hose is modeled into a coil-like shape. The remaining option is to select each row of UVs and move them one at a time to form a UV map that is tileable.
The UV problem could have been avoided by choosing a different modeling method. The cylinder could have been extruded along a straight curve, creating a long cylindrical pipe. As discussed in the concepts section of the chapter, it is much better to use the cylindrical UV mapping function before manipulating the geometry. From here the pipe is easily rigged with a simple skeleton and then positioned into a coiled shape.

Modeling and surfacing the garden hose using the polygonal extrusion method is possible, however, it can cause UVing and surfacing problems if not performed in a specific order. The advantage to following the correct order of modeling, UVing, rigging, then positioning is that the garden hose is able to be positioned into different poses. The disadvantage to following the correct polygonal creation method is the extra time spent on rigging and placing the garden hose.

Figure 7.8 The NURBS extrusion method uses a curve in the shape of a hexagon to define the shape of the extrusion.
Figure 7.9 The extruded NURBS garden hose with the red arrow texture assigned to the geometry.

The NURBS extrusion is slightly different from the polygonal extrusion in that it requires two curves: a curve to extrude along and a curve to define the shape of the extrusion. The shape of the NURBS garden hose is set by a hexagon (see figure 7.8). After the extrusion the red arrow texture is applied to surface the geometry (see figure 7.9). Right away the texture stretching problem is obvious, but not as severe as the polygonal stretching problem. Looking at the garden hose’s texture space reveals a promising texture coordinate layout.

The texture coordinates are, as expected from NURBS geometry, compacted to fit within the full zero to one texture space (see figure 7.10). This explains why the arrow appears to be stretched along the entire length of the garden hose. To remedy this, the NURBS geometry is divided into patches by selecting all the isoparms along the length of the garden hose and then using the detach surfaces function of the modeling software (see figure 7.11).
Figure 7.10 The texture coordinates for the NURBS garden hose are forced to fill the zero to one texture space.

Figure 7.11 The isoparams are selected for the detach surface function of the modeling software.
The detach surfaces function divides the garden hose into individual pieces (patches) at each isoparm. Each patch has its texture coordinates refitted into the zero to one space which results in what appears to be the arrow tiling along the length of the garden hose (see figure 7.12). In actuality the NURBS patches are separate pieces of geometry; however, they fit together seamlessly. Now that the arrows are appropriately tiling, the NURBS geometry can be converted to polygons for use in a real-time environment.

NURBS geometry cannot be displayed by most real-time rendering engines because real-time rendering engines are built to process polygons. So the NURBS garden hose is converted to polygons by using the modeling software’s convert to polygons function. The convert to polygons function does an excellent job of converting the NURBS geometry to polygons without changing the appearance of the surfacing (see figure 7.13). The resulting geometry is still a group of separate objects, so they must be combined to form a complete single polygonal garden hose.
After observing both modeling methods, the NURBS modeling process has the advantage of completely avoiding any direct manipulation of texture coordinates and saves considerable time. The disadvantage to using NURBS geometry is that it does not work as well for non-tileable textures. This is because the UVs that resulted from the NURBS-to-polygons conversion process are not sewed together. They would need to be sewed together if a non-tileable texture were to be used.

Scenario two: the air hockey table

The objective for this scenario is to analyze an air hockey table with a focus on efficient modeling, which leads to efficient surfacing. The modeling is examined from two viewpoints: from modeling the table as a single piece of geometry and from modeling the table in separations. Each method uses the same table shape (see figure 7.14) and the same textures, but each has different subdivisions in the internal geometry.
Figure 7.14 The shape of the air hockey table used by both modeling approaches.

Figure 7.15 The geometry of the non-separated air hockey table.
The non-separated geometry has the table surface, the bumper edges, and the goals connected as a single piece of geometry. Not to be confused with a grouped or combined object, the vertices are welded together. Because the table is a single piece of geometry, extra edges are needed to prevent vertices from forming t-junctions. This is why there are several edges crossing the table surface (see figure 7.15). For each edge added to a corner to increase the appearance of the curvature, extra edges must also be strung across the table surface to prevent t-junctons. This pattern continues throughout the non-separated air hockey table but is avoided in the geometry of the separated air hockey table.

The geometry of the separated air hockey table is made by combining five individual pieces of geometry: the table surface, two bumper edges, and two goals (see figures 7.16-7.18). By separating the geometry, extra detail can be added to areas like the corners without impacting the other geometry, like the table surface. All five of the pieces benefit from separation either through improved visual detail or through improved modeling efficiency.

Figure 7.16 The separated geometry of the goals.
Figure 7.17 The separated geometry of the bumper edges.

Figure 7.18 The separated geometry of the table surface.
The separated geometry modeling method can also take advantage of the air hockey table's symmetrical nature. The symmetrical nature of the table affords the use of the duplication function to improve modeling and surfacing efficiency. The duplication function perseveres the appearance and position of textures while allowing simple translations, rotations, and reflections to the geometry. As stated earlier, the air hockey table consists of two bumper edges, two goals, and a table surface. Because the bumper edge and goal are identical to its counterpart, each needs to be created only once. An artist can model and surface one goal and one bumper edge, then use the duplicate function to create a second goal and bumper edge. By using the duplication function, an artist is required to model and surface only three pieces of geometry rather than five, saving him time.

A final comparison between the geometry of the two air hockey tables reveals that the separated table uses 196 triangles and the non-separated table uses 212 triangles. A difference of 16 triangles is insignificant when viewing only the table in a real-time environment; however, if the geometry separation concept is applied to multiple objects within a scene, the polygon savings quickly add up.
Figure 7.19 A comparison between the geometry of the non-separated table (left) and the separated table (right).

Figure 7.20 A comparison between the surfaced tables. Non-separated on the left and separated on the right.
Figure 7.21 A perspective comparison between the separated table (left) and the non-separated table (right).
CHAPTER 8

MFS PROCESS SUPPOSITION

This process supposition examines an environment from Prince of Persia: The Two Thrones for its possible use of MFS. The supposition follows the construction of the environment from the perspectives of two theoretical art production pipelines: a pipeline that utilizes MFS and one that does not. The supposition is not a scientific experiment; the two pipelines were not tested and evaluated for their performance. It is a thought experiment, which demonstrates the effective benefits MFS can lend to an art production pipeline.

The artists at Ubisoft Montreal recreated the Tower of Babel and other parts of ancient Babylon (whether historically accurate or not) for Prince of Persia: The Two Thrones. What drives the authenticity of the experience is the amount of detail modeled and surfaced into the environments (see Figure 8.1).

A quick evaluation reveals a good balance between modeling and surfacing to convey details about the environment. The walls are painted with a mosaic of complex textures and the modeling is full of depth and detail (Figure 8.1). Achieving these results is a complicated task that requires the skills of talented artists working together on a production pipeline.
Area of Study

The area of study has been segmented from the overall environment (see figure 8.2), and is referred to as the "terrace." The terrace is chosen because it utilizes multiple unique textures, it can be composed of multiple geometric meshes, and it can be constructed by MFS.

Overview of the Terrace

Beneath the surfacing of the terrace walls are meshes of polygonal geometry. Polygonal walls are very similar to drywall walls found in most homes. A wall in a home is typically created by hanging multiple pieces of drywall adjacent to one another. The drywall is then patched together, sanded, and wallpapered to create the illusion of a single continuous wall. A person may know a drywall wall consists of several pieces but they cannot see the individual parts of the whole. The same is true for the walls in the terrace environment, although instead of drywall
there are polygons. The polygons are hidden from the viewer by surfacing just like wallpaper hides the seams in the drywall.

Without access to the original terrace assets, it is impossible to accurately determine the exact boundary of the polygons and their sub-divisions. The unknown sub-divisions of the geometric meshes make it difficult to predict the UV layout, or if multiple UV layers were used. For the purpose of this study, the meshes and sub-divisions of the terrace are placed based on the needs of the production pipeline in use.

The overview begins by observing the geometry of the walls. The walls are the sides of two separate buildings placed side by side (Building A and Building B; see figure 8.2). There is an
extruded column that divides the two buildings, which appears to belong to building B. Each building has windows, window frames, and ledges. The approximate scale of the walls can be determined by placing the playable character and the wall in the same frame (see figure 8.3). Observing the surface details is a much simpler task when compared to observing the geometry. This is because the textures and their separations from one another are clearly visible. A count of textures used reveals a list of 13 unique textures applied to the walls. Eleven of the textures are repeatable along U, V, or both, and two textures are not repeatable in any direction (see figure 8.4).
Figure 8.4 A list of the individual textures used within the terrace.
Production Pipelines

The construction of the terrace is viewed through two production pipelines. The first pipeline is linear and segregates the modeling and surfacing tasks. The second pipeline has the artist concurrently modeling and surfacing, allowing him to collaborate between the two specialties (or use MFS) during construction. Each pipeline is supplied with identical design documents. Both pipelines begin the construction of the terrace by evaluating the design document. The design document, prepared by art directors and concept artists, describes the purpose, uses, and look of the terrace for in-game. This document informs the artist of the scale of the terrace within the environment. It also explains which details are modeled verses surfaced (this is based on the object's importance in the scene; because the terrace is prominent within the environment, detailed areas such as windows are modeled rather than surfaced.). For this case study, figures 8.1-8.4 will act as images supplied through the design document for both pipelines.

The Non-MFS Production Pipeline

The non-MFS production pipeline is linear and segregates the modeling and surfacing tasks. The artist will begin the pipeline by modeling and finish with surfacing.

The artist, after review of the design document, begins the modeling of the terrace. The modeling is straightforward with no time spent on subdividing the geometry for surfacing or for multiple UV sets. The result is a clean and efficiently modeled terrace (see figure 8.5). The artist separates the geometry because it allows for a more efficient use of polygons (see figure 8.6). The artist then begins the surfacing process. After a review of the design document, he evaluates the model to determine the appropriate UV layout based on the texture application. After comparing the texture layout from the design document (see figure 8.4) with the model's mesh subdivision (see figure 8.5), the artist determines that the terrace requires a minimum of 10 texture files and four UV sets. The first UV set is for texture files 1-4. The second UV set is for
Figure 8.5 The geometric subdivision of the terrace from the non-MFS pipeline.

Figure 8.6 Areas of the terrace are separated for more efficient modeling.
Figure 8.7 The non-MFS pipeline could possibly require 11 textures placed within 10 texture files.

texture files 5-7. The third UV set is for texture file 8. The fourth UV set is for texture files 9 and 10 (see figure 8.7). At this point, the artist has two options: layout the UVs and continue surfacing or return the terrace to the beginning of the modeling process for modifications.

If the artist chooses to continue the UV layout process, he could run the risk of creating an inefficiently surfaced asset. The most inefficient area being poor texture file management. Out of the texture files from figure 8.7, only seven (textures 1-4, 6-7, and 10) can be reused for other areas within the game. Four of the textures (5, 8-9, 11) can only be used within the terrace. So four textures within three texture files are taking up texture memory for one wall within an entire environment. The inefficient texture management would eventually be discovered in later stages of development and returned for renovation. The artist would be forced to pass the terrace through the non-MFS pipeline a second time to make modifications to its geometry, UV layout, texture management, and texture application.
The artist is quick to realize that the terrace cannot be efficiently surfaced due to its current subdivision layout. He wants to surface the terrace using two UV sets and as few texture files as possible. After some thought, the artist recognizes that the terrace can be efficiently surfaced if it has a few polygons added by subdividing the geometry along the texture divisions (see figures 8.8-8.9). To make the changes, the artist must return the terrace to the beginning of the pipeline. This will be a time consuming maneuver, but necessary to achieve an efficient art asset.

After a look at the non-MFS production pipeline, it is apparent that communication and workflow problems exist between modeling and surfacing when tasks on the pipeline are segregated into specialties. This is due to the linear flow of work down the pipeline. The problems are significant in the scope of the entire project. The non-MFS pipeline could potentially require a remodel or modification to nearly every model created on it.

The MFS production pipeline

As stated earlier, the MFS production pipeline allows the modeling and surfacing specialties to collaborate and communicate during the construction of the terrace. In this production pipeline, the artist reviews the design document, taking notice of how and where surfacing will be integrated into the modeling process.

The artist begins modeling after reviewing the design document. During the review, the artist becomes aware of modifications to the terrace's geometry that are needed for successful surfacing. He understands that the terrace requires extra subdivisions that run along the texture edges. His modeling work generates the model in figures 8.8-8.9.

With the modeling completed, the artist can now appropriately surface the terrace. In the MFS pipeline, the terrace requires only seven texture files (see figure 8.10) and two UV sets. Tex-
Figure 8.8 The geometric subdivision of the terrace from the MFS pipeline.

Figure 8.9 Areas of the terrace are separated for more efficient modeling.
Figure 8.10 The MFS pipeline uses more textures (13), but they are packed into 7 texture files.

Textures 1-3 and 5-13 from figure 8.4 are applied to the first UV set and texture 4 from figure 8.4 is applied to the second UV set. Texture 4 is applied through a second UV set for efficiency purposes. Applying the texture through the second UV set preserves the roundness of the arch. To achieve a similar roundness through subdivision, the modeler would have to add hundreds of polygons to the terrace.

The MFS production pipeline encourages the artist to model and surface simultaneously. This concurrent development will not only satisfy the surfacing needs for the model, it will shape the terrace into a highly efficient art asset within the game. The artist's knowledge of modeling and surfacing specialties combine to produce the best modeled and surfaced asset possible.
Figure 8.11 A comparison between the geometric subdivisions of the two pipelines.
Conclusion

After following the construction of the terrace through the two theoretical production pipelines, it can be seen that the MFS pipeline is more beneficial than the non-MFS pipeline. The non-MFS pipeline encounters backtracking due to geometry modifications needed for surfacing. It requires the design document to be reviewed twice: first from the modeling perspective, then from the surfacing perspective. Finally, it runs the risk of generating inefficient assets that will be returned for remodeling and resurfacing.

One can imagine that the artist’s choice to use the non-MFS pipeline would change over the life of *Prince of Persia*’s development. The artist should be competent enough to understand the non-MFS pipeline’s shortcomings and learn from his mistakes. This assumption was true for the environment artists working on *Tiger Woods 08*. They understood the intertwined nature of modeling and surfacing and purposely developed pipelines that integrated surfacing into the modeling process.
CHAPTER 9

CONCLUSION

The observation and survey research data collected from the art production pipeline at Tiburon showed that MFS is an effective and beneficial modeling process for artists. The research also uncovered the game development process used by Tiburon, the existence of the environment artist and technical artist, and the use of MFS by the environment artist.

Rapid changes in technology that led to vast improvements to game development processes have changed the roles of artists working on real-time art production pipelines. Artists are evolving from generalists to specialists, causing their responsibilities to change. The changes in responsibilities are forming positions new to art production pipelines. While some artists are now fully specialized, some still remain in a semi-generalized state. At Tiburon, it was observed that environment artists are semi-generalized; required to know and practice multiple specialties.

The semi-generalized state is created from the need to produce highly efficient art assets. Tiburon supports the environment artist position on the pipeline because having environment modelers and surfacing artists is problematic. They found that one artist who performed both responsibilities ameliorates the problem. This arrangement has existed since the beginning of the 2000-2001 video game console generation. It is likely that Tiburon will keep the envi-
ronment artist position as semi-generalized until the demands for a higher level of art force the change to specialization. The change will also likely come about when hardware technology improves to a point where it can support a higher level of art quality with a lower level of combined modeling and surfacing efficiency without sacrificing the game's performance (This change is unlikely in the foreseeable future).

Even if improvements in hardware technology allow the environment artist position to break apart into specialties, there will be a need to exercise MFS. MFS facilitates the efficient development of art assets so that modeling and surfacing can be accomplished faster. Which results in the ability to produce art assets of a higher quality and quantity. The results of MFS can lead to real-time environments that are more realistic, engrossing, and enjoyable.

It was also observed that the new position of technical artist is playing an important role in MFS and the development of art assets. The transition from the 2000-2001 to the 2005-2006 video game consoles has brought tremendous improvements in hardware technology. Visualization techniques that were once GPU expensive (normal mapping, multiple pass renders, procedural textures, lighting and shadowing, etc.) and reserved only for special art assets are now available to all assets. Tiburon decided to remove the new technology burden from the environment artists, preferring that they concentrate on developing high quality art assets. The TA position was created to tackle the problems surrounding changes in technology and to present solutions in an approachable and usable manner.

The TAs also worked with environment artists to write shaders. Shader writing played an important role in the development of assets, not only from an aesthetic standpoint, but also from its effect on the MFS process. The environment artists pointed out that shaders would often dominate the modeling process, requiring modifications to an object's subdivisions and UV layout to fit the shader. Because of the strong impact of shaders on models, the TAs and
environment artists would collaborate to develop shaders that were more flexible and adaptable to the MFS process.

With the conclusion of this thesis, it has become evident that materials, curriculum, and learning environments related to real-time art production pipelines should include information regarding MFS. MFS plays a significant role in the development of art assets on the real-time production pipeline and, therefore, deserves attention from future authors, educators, and students. The researcher proposes the following recommendations to curriculum, course arrangements, and classroom settings.

Curriculum

After observation of the game development process and an art production pipeline at Tiburon, it is recommended that an environment art student's education cover a coherent examination of the modeling and surfacing specialties. When modeling and surfacing are taught together, the intertwined nature of modeling and surfacing becomes more apparent. While most MFS techniques can be taught in the classroom or written in books, the best learning experience for a student is through direct experience.

Learning through experience cannot be recommended any higher by the researcher. It is the best way for a student to fully grasp the concepts behind MFS. Educators can stimulate learning through experience in their course work. During the teaching of MFS concepts, an educator can assign projects that require a student to analyze models, textures, and UV sets for maximum efficiency. For example, the educator could provide a modeled, UVed, and surfaced environment object to a student. The educator would instruct the student to examine the object for its modeling and surfacing efficiency. The goal for the student would be to adjust the object's modeling and surfacing to produce a more efficient object. The educator could demonstrate the effectiveness of the student's reworked objects by rendering them in a real-time
rendering engine which displays statistics such as the frame rate, texture memory usage, and GPU performance.

It is also recommended that all student artists learn at least the basics to a widely used programming language such as C++ or JavaScript. Having knowledge of programming languages will help the student survive in the 3D industry in numerous ways. Many professional studios modify popular 3D software packages (like Maya) through plug-ins and scripts. The artist may be required to read or write some basic code to use those plug-ins and scripts. Another use for programming knowledge is for speaking with programmers and technical artists. Presenting ideas or problem solving is always easier when both parties can speak a common language.

Course arrangements

The arrangement of learning through courses can greatly impact a student's understanding and practice of 3D art creation methods. The student should avoid learning specialties one at a time. Doing so reinforces the idea of specialization and downplays the importance of inter-disciplinary techniques like MFS. This can be avoided by teaching related specialties in tandem. It is recommended that a student learn the basics to real-time art generation through the following consecutive courses:

Course 1: Modeling and Surfacing
Course 2: Rigging and Animation
Course 3: Advanced Surfacing and Lighting
Course 4: Programming Fundamentals for 3D Art Generation
Each of the first three courses should follow a similar learning track. In each course, one specialty is dependent upon the other to some extent, making the learning order critical. Learning specialties in order is important but only to an extent. It would be unrealistic to expect a student to understand complex concepts like those from MFS without first having some knowledge and practice of basic modeling and surfacing concepts. However, learning specialties apart from one another should be limited to the fundamentals necessary for complex concept understandings. It is recommended that a student spend no more than three weeks learning individual specialty fundamentals.

Classroom settings

A traditional classroom setting is an acceptable environment for teaching a student the basics to real-time 3D art generation, however, it is a poor reflection of real-world working environments. At some point in a student's education, he should be exposed to practical working simulations, like those seen at Tiburon. A student needs to experience the realities of team deadlines, inter-disciplinary communication, and team management.

It is recommended that a student have at least one course that focuses on a teamwork project to prepare a student for work on a professional art pipeline. The classroom settings should be changed to reflect the course's focus on teamwork and real-world working simulations. The classroom should be set into a pod-like configuration, grouping seating arrangements into pods of three to six. Each pod should be populated with students of similar disciplines, like character art, environment art, and animation. The duration of the class should be scheduled for three hour blocks, which simulate a typical working period on a professional pipeline.
Next steps and continued research

The scope of the research conducted during this thesis was limited to studying only one art production pipeline at the Tiburon studio of Electronic Arts. Observing the game development process at Tiburon is considerable when compared to other studios in North America. EA is the largest producer of video games in the western hemisphere, with Tiburon grossing the second largest revenue within the company. However, the sampling of Tiburon’s processes cannot be used to generalize processes at other studios.

To achieve a better understanding of the MFS process, studies must be conducted of art production pipelines at multiple studios from multiple developers at different geographic regions. By broadening the range of the sample, a researcher would have the opportunity to determine if MFS is a universal practice, or if it is only localized to Tiburon. The problem with a study of this magnitude is one of time.

The rapid and constant cycle of hardware improvements make the study of art production pipelines for real-time environments difficult. Changes in technology force changes in art production pipelines, like those seen at Tiburon. If the researcher’s study doesn’t start at the beginning of a hardware cycle change, the research could become compromised. Therefore, continued studies by a single researcher may not be the best approach.

It is the opinion of the researcher that future studies of the MFS process should be continued at multiple studios throughout North America. The studies should be conducted by individuals, then assembled as a meta-data presentation of MFS. Following this method would allow research to be conducted in a timely manner before changes in hardware technology make the meta-data obsolete.
APPENDIX A

MODELING AND SURFACING TERMS
Triangles and quads

A polygon at its most simple form is a triangle, three points on a plane. Modeling objects using triangles can be cumbersome, which is why many software programs default to modeling with quads. A quad is an extension of a triangle by a point, giving the quad a total of four vertices. Some rendering engines can process quads, triangulating them in real-time, where other rendering engines can only process triangulated geometry.

Polygon primitives

Polygon primitives are basic geometric shapes such as the sphere, cube, and cylinder. A polygon primitive can be used to generate just about any type of model. Omernick (2004) found that creating a model from primitives offers a reliable base for continued development of complex shapes( p. 19).

Polygon count

Real-time rendering engines use polygons because they are the easiest to render in a real-time environment (Omernick, 2004, p. 21). The polygon count is the total number of triangles being rendered at one time and is determined by the hardware specifications used by the rendering engine. The polygon count must be observed during the development of a real-time environment. Exceeding the polygon count can have negative consequences, such as a slow frame rate.

Polygon count management

While rendering in a real-time environment, it is important to use as few polygons as necessary. Capizzi (2002) has noted, “The determining factor for deciding how many [vertices] are required in any given piece of geometry is based on how the object looks when shaded, rendered, and animated” (p. 64). Models should be constantly evaluated for excess vertices, edges, or faces which can be removed or combined without compromising their appearance.
Clean geometry

An additional requirement to many real-time rendering engines is the use of clean geometry. A polygonal model for a real-time environment should be made entirely of triangles, or quads. Creating any of the following vertex arrangements should be avoided, as shown by Omernick (2004): overlapping faces, t-junctions, stray vertices, and n-sided polygons (p. 59-42). Overlapping faces are two faces that share part or all of the same position in 3D space. T-junctions are connected vertices that create a "T," except for border edge vertices. Imagine five vertices placed so they create a cross, with one vertex at the intersection of the cross. When the top vertex is removed, the vertices are aligned into the shape of a T. Stray vertices are extra vertices that can be deleted without affecting the shape of the geometry. N-sided polygons are polygons with more than four edges.

Normals

A rendering engine must determine which side of a polygon is facing the camera. This is determined by the polygon's face normal. The face normal is a 90-degree vector to the polygon face, and protrudes from the center of the face. The side of the polygon that the normal protrudes from is dependent upon the modeling software. Modeling software packages will determine the polygon's normal direction by analyzing the order of the vertices. The clockwise or counter-clockwise vertex order will determine the normal's protrusion direction. Capizzi (2004) has noted, "The modeler should be aware of the face normal direction, making sure that a polygon is facing the correct direction" (p. 44).

Polygon vertex, edge, and face manipulation

Vertices, edges, and faces are used to determine the surface of the model. Edges and faces can be extruded to create new polygonal faces and thus modify the shape of an object. Vertices, edges, and faces can all be moved to alter the shape of a model as well.
Texture Memory

The hardware specifications of the rendering engine will detail the amount of texture memory available for a given real-time environment. Textures must be managed to avoid exceeding the texture memory limit. The texture size is based on the texture's usage in the real-time environment (Capizzi, 2002, p 128).

Tileable Textures

Textures used in real-time environments typically fall into two categories, tileable and unique. Capizzi (2002) has written, “Tiling textures is repeating a single texture many times over a surface to increase surface detail without loading more textures into memory” (p. 129). A unique texture cannot be repeated without noticeable artifacts, but a tileable texture can.

Shaders and HLSL

According to Fernando (2003), “[A shader] makes it possible for you to control the shape, appearance, and motion of objects drawn [on screen] using programmable graphics hardware” (p. 1). Shaders are versatile because they are programmable through the HLSL (high level shading language) programming language. Fernando (2003) has also noted, “Think of a[n] [HLSL] program as a detailed recipe for how to render an object using programmable graphics hardware” (p. 2). The advancement of GPU performance coupled with talented shader writers is rapidly improving the quality of real-time environments, making shaders more important to real-time environments than ever before.

UV mapping

UVs share the same location as vertices in 3D space, however, they cannot be adjusted like a vertex. They are connected together by the same edges that connect vertices, creating a mesh. The mesh or “map” determines how a texture appears on a geometric surface. UVs are manipulated in 2D space with the U dimension along the same axis as X and the V dimension along the same axis as Y.
Multiple UV sets

An object can use more than one UV set to display its texture. Multiple UV sets are similar to painting details on separate layers, like in Photoshop. Each UV set is for different details on the surface. Typically the bottom set is reserved for tiled textures and each set above it for unique details. The UV map for each set will be laid out according to how the layer will be used.
APPENDIX B

INTERNET BASED MFS SEARCH STUDY
A survey of game development websites accompanied the printed search. Three websites were chosen for the survey: gamasutra.com, igda.org, and gamedev.org. These websites were chosen because they are reputable within the game development industry and because they allow users to search through their knowledge databases.

<table>
<thead>
<tr>
<th>Query Entry</th>
<th>Gamasutra results (returned links)</th>
<th>IGDA results (returned links)</th>
<th>Gamedev results (returned links)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Modeling for surfacing&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;Modeling for texturing&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Art+pipeline</td>
<td>78</td>
<td>9</td>
<td>148</td>
</tr>
<tr>
<td>Modeling+pipeline</td>
<td>57</td>
<td>9</td>
<td>71</td>
</tr>
<tr>
<td>Modeling+communication</td>
<td>70</td>
<td>5</td>
<td>50</td>
</tr>
<tr>
<td>Modeling+communication+texturing</td>
<td>37</td>
<td>3</td>
<td>57</td>
</tr>
<tr>
<td>Art+communication</td>
<td>125</td>
<td>60</td>
<td>162</td>
</tr>
</tbody>
</table>

Figure B.1 displays the results of queries into a website’s knowledge base (2-24-06). The units represent the number of returned pages by the search engine. The query subtracted the words “job” and “asset” to focus the results.

<table>
<thead>
<tr>
<th>Query Entry</th>
<th>Gamasutra results (relevant links)</th>
<th>IGDA results (relevant links)</th>
<th>Gamedev results (relevant links)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Modeling for surfacing&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>&quot;Modeling for texturing&quot;</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Art+pipeline</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Modeling+pipeline</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modeling+communication</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Modeling+communication+texturing</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Art+communication</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure B.2 displays the number of relevant pages to the query entry. The pages represent the number of pages judged to be relevant to the query by the researcher.
Each term in the query field of figures B.1 and B.2 was entered exactly as is into the searchable database of each website. The returned links field for each website represents the number of links to articles that were returned. Figure B.1 shows the number of links returned from each website related to each query. Figure B.2 shows the number of links from each query that the researcher believed to be relevant to MFS on the art production pipeline.
APPENDIX C
SURVEYS
**Instructions**

This survey is intended for **environment/object modelers**.

Please answer as many questions as possible. Feel free to skip any questions which are not applicable or you do not wish to answer.

Please return the survey to Joe Harmon when you are finished. Please notify Joe Harmon by email if you were unable to return the survey.

If you have any questions during the survey, please contact Joe Harmon by phone or email.

Joe Harmon
614-598-2682
jharmon@accad.osu.edu

Please answer the following questions as completely as possible.

1. What is your name and position title?
   
   Jason Alexander - Environment Artist

2. Please briefly describe your responsibilities.

   3D creation of "game based" environments for current and next gen platforms. The process of which includes, and is not limited to, modeling, lighting, texturing, UVing and conceptual development of game based assets.

3a. Briefly describe the flow of content within your current art production pipeline. (ex. modeling > surfacing > lighting > etc.)

```
concept --> modeling --> UV layout --> lighting
  research    texture creation
```

3b. Describe the modeling to surfacing process within your current art pipeline.

   Model is created first followed by surfacing on the whole. Both processes are carried out by the artist and development of the final product can involve concurrent development.

4. Do you encounter problems in the modeling to surfacing handoff?

   If yes, please describe the most common problems.
   Not really. As mentioned above, both processes are carried out by this artist in the development process.
Question 4 continued...

If no, why not?

5. How often do you collaborate with a surfacing artist to determine the modeling approach?

never  sometimes  often  always

6. Do you find collaboration helpful in the modeling process?

If yes, why?
Absolutely. Maya/3D packages have numerous permutations in the process of achieving a specific product. Different approaches may result in the streamlining of a process which can improve efficiency and overall end product.

If no why?

7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

never  sometimes  often  always

8. Are you encouraged to collaborate with surfacing artists without involving your project manager?

If yes, has it ever caused unforeseen problems?
Not as yet, as long as art direction is adhered to. It is the responsibility of the artist to ensure this.

If no, how do you communicate with the surfacing artists?
9. Is there collaboration between modelers and surfacing artists before modeling begins?
   
   If yes, what do you discuss?
   Refer to previous.

   Why is it helpful?

10. Do you receive a design document before you begin modeling?

    Yes  No

11. How does the design document aid you with modeling?
    Provides a baseline for the dev. process and hopefully prevents needless revisions of work.

12. Does the design document aid you with determining which details to model and which details to pass to the surfacing artist?

    Yes

    If no, how are the details resolved?
    Not always. Depending on complexity of the final product, not all details can be included in documentation. Again, it is incumbent on the artist to determine style and level of detail that is outlined by art direction

13. Have you ever strayed from the design document to design a better object?

    If yes, how did that effect the object's movement down the production pipeline?
    Yes. The current work environment encourages design document changes that improve workflow. Improvement in speed and work quality allowed time to be spent on other areas that needed attention.
14. How often do you remodel an object at the request of a surfacing artist?

never  sometimes  often  always

15. Have you noticed any patterns within the request for a remodel? What are those patterns?
   Usually it involves model optimizations to accommodate UV and texture layout – occasionally it has been because of object/skeletal weighting optimizations.

Who is responsible for the UV layout of objects?

If yourself, continue with the survey. If someone else, go to page 6.

16a. At what point in the modeling process of an organic object is the UV layout considered?
   Usually development is concurrent.

16b. Is the UV layout of an organic object considered more complex than the UV layout of a planar object?
   Usually – Layout complexity increases as number of axes the object is built in increases.

16c. Can the complexity of the UV layout of an organic object effect/change the modeling approach for that object?

   Yes  No
16d. Do you consider the UV layout of an object before modeling?

If yes, how?
UV layout/texture creation will determine/influence tessellation complexity in how the object is viewed. This ultimately is dependent on if detail will be modeled or textured.

If no, why?

16e. Are you in the practice of laying out the UV's of an object during the modeling process?

If yes, why?
Yes – Iterative development from my perspective usually improves speed and aesthetic output.

If no, why?

16f. Can the layout of the UV's of an object after modeling be difficult or time consuming?

[ ] Yes  [ ] No

Please continue to question 18.
If someone else is responsible for the UV layout of objects:

17a. Does thinking about UV layout influence your modeling approach?

   If yes, how?

   If no, why?

17b. Does your manager require you to model an object with the UV layout in mind?

   Yes    No

17c. How does modeling an object with the UV layout in mind effect the movement of an object down the art production pipeline?

Please continue to question 18.

18. What are the general problems with the UV layout when a remodel is requested?

   Depending on the amount of remodel required it can add significant time to the production process of object creation.
19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

Yes  No

If Yes, why?
Based on personal experience – especially if proper guidelines are adhered to.

20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

Yes  No

21. In your opinion, what surfacing knowledge must a modeler posses to be able to model objects for an art production pipeline?

Next and current gen pipelines require a slightly different skill set but lighting, normal map, bump, and texture optimization would be required to provide a good product.

22. In your opinion, what modeling knowledge must a surfacing artist posses to be able to surface objects for an art production pipeline?

Know how to optimize geometry to best suit the object.

23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.

A concurrent and iterative approach to modeling and surfacing has produced the best results from a personal perspective. It allows for a progressive review and reduced amount of rework that needs to be done on the final product.
Instructions

This survey is intended for environment/object modelers.

Please answer as many questions as possible. Feel free to skip any questions which are not applicable or you do not wish to answer.

Please return the survey to Joe Harmon when you are finished. Please notify Joe Harmon by email if you were unable to return the survey.

If you have any questions during the survey, please contact Joe Harmon by phone or email.

Joe Harmon
614-598-2682
jharmon@accad.osu.edu

Please answer the following questions as completely as possible.

1. What is your name and position title?
   Jay Anderson – Environment modeler

2. Please briefly describe your responsibilities.
   Modeling and texturing of environments which includes terrains, props, buildings, vehicles and vegetation.

3a. Briefly describe the flow of content within your current art production pipeline. (ex. modeling > surfacing > lighting > etc.)
   Modeling → Textures → Rigging → Lighting

3b. Describe the modeling to surfacing process within your current art pipeline.
   Most of the time for the last couple games I’ve been on, the modeler both models and textures.

4. Do you encounter problems in the modeling to surfacing handoff?
   If yes, please describe the most common problems.
   No
Question 4 continued...

If no, why not?
Usually after modeling, I will continue on to the surfacing.

5. How often do you collaborate with a surfacing artist to determine the modeling approach?

- never
- sometimes
- often
- always

6. Do you find collaboration helpful in the modeling process?

If yes, why?

If no why?

7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

- never
- sometimes
- often
- always

8. Are you encouraged to collaborate with surfacing artists without involving your project manager?

If yes, has it ever caused unforeseen problems?

If no, how do you communicate with the surfacing artists?
9. Is there collaboration between modelers and surfacing artists before modeling begins?
   
   If yes, what do you discuss?
   
   Why is it helpful?

10. Do you receive a design document before you begin modeling?
    
    Yes  No

11. How does the design document aid you with modeling?
    
    In most cases it will often give guidelines on how best to model the object to fit the pipeline.
    Ways to avoid possible pitfalls, as well as specs for things such as poly count etc.

12. Does the design document aid you with determining which details to model and which
details to pass to the surfacing artist?
    
    Yes
    
    If no, how are the details resolved?

13. Have you ever strayed from the design document to design a better object?
    
    If yes, how did that effect the object’s movement down the production pipeline?
    Most of the time if I plan to stray from the design document and feel it may have possible impact
down the pipeline I will check with TA's, programmers, etc to try to determine possible drawbacks.
14. How often do you remodel an object at the request of a surfacing artist?

never  sometimes  often  always

15. Have you noticed any patterns within the request for a remodel? What are those patterns?

Who is responsible for the UV layout of objects?

If yourself, continue with the survey. If someone else, go to page 6.

16a. At what point in the modeling process of an organic object is the UV layout considered?
   For me, even before I begin modeling, UV layout is being considered and I continue to adjust my plans for the layout as the model progresses.

16b. Is the UV layout of an organic object considered more complex than the UV layout of a planar object?
   Yes, just due to the overall nature of an organic object. It often takes more time and care in layout of the UVs. While a particular object is a fairly straightforward process.

16c. Can the complexity of the UV layout of an organic object effect/change the modeling approach for that object?

   Yes  No
16d. Do you consider the UV layout of an object before modeling?

If yes, how?
I often think about how to build an object to best or most effectively use the UV space. I consider whether or not a detail can be textured or modeled and of which method will be most economical for in game use. (bang for the buck)

If no, why?

16e. Are you in the practice of laying out the UV’s of an object during the modeling process?

If yes, why?

If no, why?
So often, UV’s will change through the modeling process that often I find it better to make sure a model is finalized and approved before laying out UVs.

16f. Can the layout of the UV’s of an object after modeling be difficult or time consuming?

Yes  No  Sometimes.

Please continue to question 18.
If someone else is responsible for the UV layout of objects:

17a. Does thinking about UV layout influence your modeling approach?

   If yes, how?
   Yes

   If no, why?

17b. Does your manager require you to model an object with the UV layout in mind?

   Yes  [ ]  No  [x]

17c. How does modeling an object with the UV layout in mind effect the movement of an object down the art production pipeline?

   Yes. Whether I model and texture an object or model the object and give it to somebody else. I try to model the object to make UVing the object easy as possible and to take advantage of both the texture details and details in the model geometry.

Please continue to question 18.

18. What are the general problems with the UV layout when a remodel is requested?

   Often, but not always. Depending on the changes that have to be made to the model. The UVs get trashed – usually when a remodel takes place, UVs will have to be redone. At the very least adjusted – not to mention textures that may need re-worked as well.
19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

   Yes  [ ]  No  [ ]

   If Yes, why?

20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

   Yes  [ ]  No  [ ]

21. In your opinion, what surfacing knowledge must a modeler possess to be able to model objects for an art production pipeline?

   A modeler needs at least a pretty good knowledge of how UVs are created and textures applied. In order to create models that are able to make the most of the textures they will use as well as to make the most of UV layout.

22. In your opinion, what modeling knowledge must a surfacing artist possess to be able to surface objects for an art production pipeline?

   I think that having a knowledge of how a model is created is vital. It’s better if a texture artist can model as well as texture but even if they can’t at least being aware of the processes, tools, and procedures involved is important.

23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.

   In the next gen systems I have seen a merging of the modeler responsibilities and texture artist’s responsibilities. Another addition is the importance of shaders and shader writers (TAs). I will often collaborate with a TA to plan and create a shader that will be able to work with the object that I am modeling. I will often use the knowledge of how a particular shader works in order to plan on how I will model, layout UVs, and texture an object. So much of my plan on the creation of a complicated model depends on the way I plan to use a particular shader. I have often reworked a model or had to rework a model if a shader is updated or as to change for some reason. Those changes are necessary to optimize the model but can often be very time consuming.
Instructions

This survey is intended for environment/object modelers.

Please answer as many questions as possible. Feel free to skip any questions which are not applicable or you do not wish to answer.

Please return the survey to Joe Harmon when you are finished. Please notify Joe Harmon by email if you were unable to return the survey.

If you have any questions during the survey, please contact Joe Harmon by phone or email.

Joe Harmon
614-598-2682
jharmon@accad.osu.edu

Please answer the following questions as completely as possible.

1. What is your name and position title?
   Ron Amador – Staff Environment Lead

2. Please briefly describe your responsibilities.
   To communicate, task, and prioritize information up to art direction and management and down to team artists from management.

3a. Briefly describe the flow of content within your current art production pipeline.
   (ex. modeling > surfacing > lighting > etc.)
   Model → Texture → Shader Tuning ↔ Game QA

3b. Describe the modeling to surfacing process within your current art pipeline.
   We study reference photographs and model closely large points of interest. We also look at concept art to make sure we hit on the art director's priorities. We then tune shaders and deliver assets to the lighting team.

4. Do you encounter problems in the modeling to surfacing handoff?
   If yes, please describe the most common problems.
   No. We don’t have “surfacers.” Modelers do surfacing work.
Question 4 continued...

If no, why not?
We eliminate potential issues by having modelers that are also skilled surfacers.

5. How often do you collaborate with a surfacing artist to determine the modeling approach?

- never
- sometimes
- often
- always

6. Do you find collaboration helpful in the modeling process?

If yes, why?
I feel privileged to work in an environment with lots of talented people. All of which have great ideas, and the ability to give good feedback.

If no why?

7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

- never
- sometimes
- often
- always

8. Are you encouraged to collaborate with surfacing artists without involving your project manager?

If yes, has it ever caused unforeseen problems?

If no, how do you communicate with the surfacing artists?
9. Is there collaboration between modelers and surfacing artists before modeling begins?
   If yes, what do you discuss?

   Why is it helpful?

10. Do you receive a design document before you begin modeling?
    
    Yes  No

11. How does the design document aid you with modeling?
    It doesn't aid us much. It outlines the priorities for the game. In most cases for me, it's to make as realistic and accurate environment as possible.

12. Does the design document aid you with determining which details to model and which details to pass to the surfacing artist?
    Yes

    If no, how are the details resolved?
    N/A on surfacing. Detail issues are resolved by modeling samples during our R&D phase. These are OK'd by the art director and we use it as a “key.”

13. Have you ever strayed from the design document to design a better object?
    If yes, how did that effect the object’s movement down the production pipeline?
    No.
14. How often do you remodel an object at the request of a surfacing artist?

- never
- sometimes
- often
- always

15. Have you noticed any patterns within the request for a remodel? What are those patterns?

Who is responsible for the UV layout of objects?

If yourself, continue with the survey. If someone else, go to page 6.

16a. At what point in the modeling process of an organic object is the UV layout considered?

- After modeling is complete.

16b. Is the UV layout of an organic object considered more complex than the UV layout of a planar object?

- More potential for seaming and flow issues. Plus animation is more likely on organics.

16c. Can the complexity of the UV layout of an organic object effect/change the modeling approach for that object?

- Yes
- No
16d. Do you consider the UV layout of an object before modeling?

If yes, how?
More experienced artists can develop a plan for their UV layouts while they model.

If no, why?

16e. Are you in the practice of laying out the UV's of an object during the modeling process?

If yes, why?
Sometimes if part of an object is repeated, (windows, columns) it best to layout UVs once and then duplicate.

If no, why?

16f. Can the layout of the UV's of an object after modeling be difficult or time consuming?

[ ] Yes  [ ] No

Please continue to question 18.
If someone else is responsible for the UV layout of objects:

17a. Does thinking about UV layout influence your modeling approach?

    If yes, how?

    If no, why?

17b. Does your manager require you to model an object with the UV layout in mind?

    Yes    No

17c. How does modeling an object with the UV layout in mind effect the movement of an object down the art production pipeline?

Please continue to question 18.

18. What are the general problems with the UV layout when a remodel is requested?

    If the object's shape changes, then we are forced to change UVs. So in most cases, the areas of remodel need re-UVing.
19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

Yes   No

If Yes, why?
Depends on what you mean by "problems." More experienced artists can foresee issues at every step of the process and learn to avoid them. Not to mention veterans have more experience with tools, proprietary scripts, and the people who facilitate the artists work.

20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

Yes   No

21. In your opinion, what surfacing knowledge must a modeler possess to be able to model objects for an art production pipeline?
Modelers need to know everything about the surfacing process. In cases where its (surfacing) is done by another artist, its a huge help when designing the model.

22. In your opinion, what modeling knowledge must a surfacing artist possess to be able to surface objects for an art production pipeline?
Little to none. If a "surfer" is used as part of a pipeline, his skills should mostly be traditional (illustration, paint, sculpture).

23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.
In real-time games, the idea of a surfer or a texturer is an antiquated one. In rare cases, texturers are hired at EA to texture characters.
Instructions

This survey is intended for environment/object modelers.

Please answer as many questions as possible. Feel free to skip any questions which are not applicable or you do not wish to answer.

Please return the survey to Joe Harmon when you are finished. Please notify Joe Harmon by email if you were unable to return the survey.

If you have any questions during the survey, please contact Joe Harmon by phone or email.

Joe Harmon
614-598-2682
jharmon@accad.osu.edu

Please answer the following questions as completely as possible.

1. What is your name and position title?
   Jeff Martin – Environment Modeler

2. Please briefly describe your responsibilities.
   Modeling, texturing, lighting, finding faster and more efficient ways to model and improve the pipeline. Documenting new processes.

3a. Briefly describe the flow of content within your current art production pipeline.
   (ex. modeling > surfacing > lighting > etc.)
   Modeling → Texturing → Lighting

3b. Describe the modeling to surfacing process within your current art pipeline.

4. Do you encounter problems in the modeling to surfacing handoff?
   If yes, please describe the most common problems.
   No
Question 4 continued...

If no, why not?
There never seems to be a problem. If someone else is texturing it, they will 90% of the time layout their own UVs.

5. How often do you collaborate with a surfacing artist to determine the modeling approach?

never sometimes often always

6. Do you find collaboration helpful in the modeling process?
If yes, why?

If no why?
I never talk to the texture artist about modeling. Everyone has their strength, if you’re a texture artist, your strength is texturing, not modeling.

7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

never sometimes often always

8. Are you encouraged to collaborate with surfacing artists without involving your project manager?

If yes, has it ever caused unforeseen problems?
No

If no, how do you communicate with the surfacing artists?
9. Is there collaboration between modelers and surfacing artists before modeling begins?

   If yes, what do you discuss?

   Why is it helpful?

10. Do you receive a design document before you begin modeling?

    Yes  No
    I’m usually the one writing them.

11. How does the design document aid you with modeling?

12. Does the design document aid you with determining which details to model and which
details to pass to the surfacing artist?

    Yes

    If no, how are the details resolved?
    In time and experience you will learn what should be modeled and what should be surfaced

13. Have you ever strayed from the design document to design a better object?

    If yes, how did that effect the object’s movement down the production pipeline?
    Yes. I remodeled something more efficiently and freed up memory on the PS2. Always keep in mind
    what the camera is going to see.
14. How often do you remodel an object at the request of a surfacing artist?

- never
- sometimes
- often
- always
- Almost never

15. Have you noticed any patterns within the request for a remodel? What are those patterns?

No

Who is responsible for the UV layout of objects?

- **If yourself, continue with the survey. If someone else, go to page 6.**
- 75% sometimes myself, 25% sometimes someone else

16a. At what point in the modeling process of an organic object is the UV layout considered?

When I’m finished modeling

16b. Is the UV layout of an organic object considered more complex than the UV layout of a planar object?

Planar is easy. A complex object like a human will take cylindrical, planar, and automatic mapping.

16c. Can the complexity of the UV layout of an organic object affect/change the modeling approach for that object?

- Yes
- No
16d. Do you consider the UV layout of an object before modeling?
   
   If yes, how?

   If no, why?
   No, worry about that when I'm done modeling.

16e. Are you in the practice of laying out the UV's of an object during the modeling process?

   If yes, why?

   If no, why?
   Wait till the model is completed and has the proper OK by upper management, it may change a bit.

16f. Can the layout of the UV's of an object after modeling be difficult or time consuming?

   Yes ☐ No ☐

Please continue to question 18.
If someone else is responsible for the UV layout of objects:

17a. Does thinking about UV layout influence your modeling approach?
   
   If yes, how?
   
   If no, why?
   No

17b. Does your manager require you to model an object with the UV layout in mind?
   
   Yes  No

17c. How does modeling an object with the UV layout in mind affect the movement of an object down the art production pipeline?

Please continue to question 18.

18. What are the general problems with the UV layout when a remodel is requested?
   The border edges may get moved around and screw up your UV layout.
19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

Yes  No

If Yes, why?
Sometimes, they don't understand to use all the UV space (0 to 1) and give a little object more resolution than the main object that needs it.

20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

Yes  No

21. In your opinion, what surfacing knowledge must a modeler possess to be able to model objects for an art production pipeline?

Good eye for detail.

22. In your opinion, what modeling knowledge must a surfacing artist possess to be able to surface objects for an art production pipeline?

Good eye for detail.

23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.

I like to keep everything in quads as much as possible. It's cleaner and easy to reduce or take out edges!
Instructions

This survey is intended for environment/object modelers.

Please answer as many questions as possible. Feel free to skip any questions which are not applicable or you do not wish to answer.

Please return the survey to Joe Harmon when you are finished. Please notify Joe Harmon by email if you were unable to return the survey.

If you have any questions during the survey, please contact Joe Harmon by phone or email.

Joe Harmon
614-598-2682
jharmon@accad.osu.edu

Please answer the following questions as completely as possible.

1. What is your name and position title?
   David Allen Wade
   Associate Environment Modeler

2. Please briefly describe your responsibilities.
   To create assets related to environments, or to model, texture, shade any asset that isn’t a character.

3a. Briefly describe the flow of content within your current art production pipeline.
    (ex. modeling > surfacing > lighting > etc.)
    Pre-vis —> Concept —> Modeling —> Texturing —> Lighting/Shading

3b. Describe the modeling to surfacing process within your current art pipeline.
    Create geometry —> Layout UVs —> Normalize surface —> Create file (map) textures —>
    Shading/Lighting —> Profit

4. Do you encounter problems in the modeling to surfacing handoff?

   If yes, please describe the most common problems.
   Generally, both are handled by the same artist.
Question 4 continued...

If no, why not?

5. How often do you collaborate with a surfacing artist to determine the modeling approach?

never  sometimes  often  always
Rarely

6. Do you find collaboration helpful in the modeling process?

If yes, why?

If no why?

Many modelers have different approaches and levels of competency.

7. How often do you remodel an object because of problems, miscommunications, or changes by surfacing needs?

never  sometimes  often  always
Mainly due to producers.

8. Are you encouraged to collaborate with surfacing artists without involving your project manager?

If yes, has it ever caused unforeseen problems?
No, not really.

If no, how do you communicate with the surfacing artists?
9. Is there collaboration between modelers and surfacing artists before modeling begins?

If yes, what do you discuss?
Environment artists aren’t typically segmented into these groups.

Why is it helpful?

10. Do you receive a design document before you begin modeling?

Yes    No
Sometimes

11. How does the design document aid you with modeling?

Design documents typically provide links to reference and specs (like file size or poly count) on an asset. However, they are usually available only after an asset type has already been established.

12. Does the design document aid you with determining which details to model and which details to pass to the surfacing artist?

Yes

If no, how are the details resolved?
By artist interpretation or by art director consultation.

13. Have you ever strayed from the design document to design a better object?

If yes, how did that effect the object’s movement down the production pipeline?
It was met by furious opposition from makers of previous versions, and it went unnoticed by everyone else.
14. How often do you remodel an object at the request of a surfacing artist?

never  sometimes  often  always

15. Have you noticed any patterns within the request for a remodel? What are those patterns?

Just one. A lack of artistic education.

Who is responsible for the UV layout of objects?

If yourself, continue with the survey. If someone else, go to page 6.

16a. At what point in the modeling process of an organic object is the UV layout considered?

After geometry is created, before normals are considered.

16b. Is the UV layout of an organic object considered more complex than the UV layout of a planar object?

Naturally, yes. An organic object would require more consideration in the UV mapping process (seams are more visible on an organic object), as well as the UV layout process (organic objects tend to deform more – requiring a layout that is tailored toward its deformation).

16c. Can the complexity of the UV layout of an organic object effect/change the modeling approach for that object?

Yes  No
16d. Do you consider the UV layout of an object before modeling?

If yes, how?
Indirectly. When modeling I'm considering the shape and interaction of surfaces. So, I'm naturally going to visualize those surfaces in UV form while I'm modeling.

If no, why?

16e. Are you in the practice of laying out the UV's of an object during the modeling process?

If yes, why?

If no, why?
Surface geometry changes too frequently. UV layout has to happen after modeling is completed for sake of efficiency.

16f. Can the layout of the UV's of an object after modeling be difficult or time consuming?

Yes  [ ] No  [x]
Not if you know what you're doing :)

Please continue to question 18.
If someone else is responsible for the UV layout of objects:

17a. Does thinking about UV layout influence your modeling approach?
   If yes, how?
   If no, why?

17b. Does your manager require you to model an object with the UV layout in mind?
   Yes    No

17c. How does modeling an object with the UV layout in mind effect the movement of an object down the art production pipeline?

Please continue to question 18.

18. What are the general problems with the UV layout when a remodel is requested?
   If changes in geometry affect UV seaming, or if additional geometry warrants a change in layout to accommodate space, then it's a problem.
19. Do modeling interns or new hires with little to no experience working within an art production pipeline have problems moving work down the art production pipeline?

Yes  No

If Yes, why?

20. Would a company benefit from hiring artists that had a working knowledge of art production pipelines?

Yes  No

21. In your opinion, what surfacing knowledge must a modeler posses to be able to model objects for an art production pipeline?

They must know the constraints of texturing in order to produce a usable UV layout.

22. In your opinion, what modeling knowledge must a surfacing artist posses to be able to surface objects for an art production pipeline?

Little or none. They only would need to see the model in a 3D space to understand the correspondence between the UVs and geometry.

23. Please use the following space to share your personal experiences with the modeling to surfacing process, or any other information you believe to be helpful.

Hmm... Automatic mappings an utterly useless process in Maya. Also, there's been no mention of topology in this survey. Topology is a rather critical factor in determining geometry and layout.
LIST OF REFERENCES


ANNOTATED BIBLIOGRAPHY


This book begins by tracking the history of architectural renderings, starting with the introduction of perspective in the 15th century. Chapter six gives good details about virtual worlds and their make up. The author describes the elements that are needed to create a virtual world and how they should be used. The author also writes about control devices to interact with virtual worlds and their effects on the five senses. The author concludes the book with current (at the time) and future research.

This book is a good resource for those interested in using virtual worlds for architectural space. It gives many clear examples of the effective uses for virtual worlds, listing many pros and cons. The book provides lots of theory but little process information. This book is useful for those interested in learning about virtual worlds and useless for those interested in building virtual worlds.


Bethke is the executive producer of the Starfleet Command series and Black9. Bethke has worked his way through the video game industry to his current position as CEO of Taldren, and uses this book to share his experience and knowledge about game development. Bethke covers the entire spectrum of the production pipeline, from idea inception to post-release. The majority of the book is written from the perspective of a game designer looking over the production pipeline as an idea turns into a product.

Bethke's book is intended to aid game designers, producers, and directors. Most of the information is presented from those perspectives, a top down view of the production pipeline. He spends a lot of time writing about what happens before assets hit the art production pipeline, which makes sense if you are writing to game designers, producers, and directors. Unfortunately, Bethke's explanation of the art production pipeline is nothing more than descriptions of jobs and their responsibilities. He gives the art production pipeline little attention compared to the chapters on game concept, vision document, and game design.
Ed Byrne is a game designer and level designer who has worked on games such as *Harry Potter and the Prisoner of Azkaban* and *Splinter Cell*. Byrne describes his book as "...the fundamentals of level design: it teaches you common procedures for designing, drafting, and creating interactive environments for games." The majority of the book describes the role of a level designer for a video game while working inside of the art production pipeline.

Byrne's book provides insight to the collaboration between artists that brings a level to life. He explains the role of level designer as a compiler of art assets. The level designer is responsible for taking the art from modelers, surfing artists, and animators and placing it into a game level. The level designer also scripts events the player encounters throughout a level.

*Game Level Design* is a great resource for those interested in the challenging role of level designer. It also grants some brief exposure to the communication between modelers and surfing artists. Byrne tells us that modelers and surfing artists communicate to make designing the level easier (p. 249-250). Unfortunately Byrne leaves much unanswered about modeling for surfacing communication.


Capizzi has been working at Rhythm & Hues Studios since 1996, working on several major movies such as *Dr. Dolittle 2*, *The Flintstones in Viva Rock Vegas*, and *Spawn*. His book covers the role of modeler and surfing artist for animations and real-time productions. Capizzi also interviews professionals from other studios on their experiences with CG.

Modeling and surfing are presented as separate jobs on the production pipeline throughout the book. However, Capizzi explains some of the subtleties of the modeling/surfacing artist relationship. This mostly deals with the UV layout process.


This book reads very much like a technical manual for virtual reality and the human body. It goes into an overwhelming amount of detail regarding the physiological and psychological effects of virtual reality and simulated reality on humans.

Chapter three of this book was particularly interesting with its discussion on virtual reality and human perception. The author presented interesting information regarding the physical capacities of human eyes and the technical limitations of computer monitors. The author was clear to point out that display technology does not match human perception. It cannot fully display
what we can fully see. However, this is nothing new. The author points out that limitations in mediums have led to short cuts in realism throughout history. The book looks to painting and the technique of shifting colors to help convey emotion, ideas, time of day, etc. The author says the problem of realism lies with compression. Realism must be compressed to fit within a mediums limitations.


The Game Asset Pipeline is a book for those interested in understanding the process of asset management in a video game production pipeline. Carter writes a comprehensive guide for what it takes to compile game assets from source code to playable disc. Carter provides asset management advice that will allow development teams to make progress and avoid the consequences of mismanaged files.

This book explains how hardware and software combine to present a 3D world in a real-time environment. It does not shed light on the art production pipeline. It is, however, an interesting read for those unfamiliar with the processes that work behind the computer screen.


This book is written with the intention of giving the reader an understanding of how to decipher and describe what they see and to focus on the process of creating textures. Where an artist would find this book very useful, a designer is left wanting. Too much time was directed towards art generation rather than design examples. This would be a good read for someone with no experience creating or implementing textures in a 3D program.


This is the book for learning about programmable shaders. It starts out at the very basic level of explaining GPUs, their history, and how the CG and HLSL languages evolved. From there the book dives into coded examples, most of which can be practiced by the reader. The only downside to the book is that some of the examples must be tweaked by the reader because of changes in Nvidia’s FX Composer software. This is a must-read for all aspiring technical artists.


Todd Gantzler has a long history with teaching 2D and 3D computer art. He is currently the program leader for the Computer and Video Games honors degree program at the University
of Salford in England. Gantzler wrote this book as a classroom teaching aid, and includes many helpful visual aids and explanatory text. His book covers only modeling and surfacing, but does so at a deeper level than many other books of this genre.

Gantzler is one of the few authors who have acknowledged the symbiotic relationship between modeling and surfacing. In his preface he says, “Texture mapping is often taught after modeling; however, modeling and texture mapping are dependant on one another to some extent.” He covers modeling and surfacing in the traditional manner, separately, but he does explain where and when the two become codependent.


On page 56 Kerlow diagrams a potential production pipeline for a small team working on an animated short.


Riccard Linde has 10 years of experience working in the video game industry. He has worked on Battlefield 1942, Road to Rome, and Secret Weapons of WW2. He has held positions of director, lead artist, and art technical. Linde's book is a learning resource that explains the workings of the art production pipeline for real-time video games. He explains the knowledge and techniques a student needs to know to land a job in video games.

Linde does an excellent job of explaining how the video game industry is becoming specialized and how the specialties interact with one another. This perspective is important; students need to understand how game development for the current generation of consoles will differ from the next generation of consoles. His chapters on modeling, UV mapping, and surfacing are beyond instructive. Linde explains not only the "what" and "how" about techniques but also the "why." He shows why techniques overlap between specialties and explains the reasoning and benefits for the overlap.

Linde also highlights the technical information behind the art techniques. This information is very helpful for building efficient models or creating streamlined art.

On page 82-83 Linde describes the modeling process of modeling for UV reuse. He explains how an object's geometry is sub-divided for texture application. An illustration on page 83 shows a polygonal building with many of the polygons colored orange and green. Linde explains that the colored polygons represent tiled textures. The colored polygons are arranged to
break up the texture on the building to give it a more realistic look. On page 82, Linde explains how extra polygons can be spent to improve the texturing of the model: "... you can create extra polygons around doors, windows, and the corners of buildings to break the uniformity of the tile [tiling texture] by applying different texture variations."


Matthew Omernick is an experienced game artist/designer, working for Electronic Arts and LucasArts throughout his career. He brings this experience to the book, leading the reader through the video game design pipeline. Detailing the pipeline is the most useful information presented in the book. He gives good suggestions about making large projects workable and tips for each stage of the pipeline.

When modeling in a real-time environment, it is necessary to use as few polygons as possible. Omernick doesn’t call it “low-poly” modeling but rather “creative poly” modeling. This is very true. Knowing where to save polygons and where to exploit them is very important.

He presents examples of modeling, texturing, lighting, and special effects in games with a good overview to get one started in the game industry. However, this book would not be an excellent resource for one needing troubleshooting help with their designs.


Randy Pausch is a professor of Computer Science, HCI, and Design and the co-director of the Entertainment Technology Center at Carnegie Mellon University. Pausch held a residency at Electronic Arts (EA) during the spring of 2004 to observe and report on EA and to prepare students for work within EAs studios. The reasoning behind Pausch’s residency could be summed up by his statement: "It immediately became clear to me that neither EA nor academia have any real understanding of how the other operates."

Pausch speaks mostly about surviving the corporate climate at EA. He says that students must go to EA with an outstanding work ethic or risk losing their jobs. He says that one of the integral keys to EAs success is the quality of their management. Apparently, EAs managers are very effective in extracting the best work from their employees while keeping a game on schedule.

Pausch reports that "EA has as many pipelines as projects, and they are highly ad-hoc and painful to maintain." He also found that EAs art pipelines are "highly variable across projects and sometimes even within a project." The report says that this is due to rapidly changing local circumstances. He also found that EA does little to no research and development or publication of papers. If EA holds this attitude towards research, it could be held by other American developers, which could explain the lack of detailed information on the art production pipeline.