HIGH NOON: DEVELOPING A VIDEO GAME PRODUCTION PIPELINE FOR CHICO STATE GAME STUDIOS

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

In Partial Fulfillment
of the Requirement for the Degree
Master of Science
in
Interdisciplinary Studies
Applied Computer Graphics

by
© Alisha Thayer 2009
Summer 2009
HIGH NOON: DEVELOPING A VIDEO GAME PRODUCTION PIPELINE FOR CHICO STATE GAME STUDIOS

A Project

by

Alisha Lynn Thayer

Summer 2009

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

_________________________________
Mark J. Morlock, Ph.D.

APPROVED BY THE GRADUATE ADVISORY COMMITTEE:

Sara A. Trechter, Ph.D.  Clarke H. Steinback, Ph. D., Chair
Graduate Coordinator

Byron Wolfe, M.F.A.

Frank Armstrong, M.F.A.
PUBLICATION RIGHTS

No portion of this project may be reprinted or reproduced in any manner unacceptable to the usual copyright restrictions without the written permission of the author.
ACKNOWLEDGMENTS

I would like to formally thank my committee, Clarke Steinback, Byron Wolfe, Sara Trechter, and Frank Armstrong for their help and support during this project. Your enthusiasm and interest during my time in graduate school meant more to me than you can know.

My sincerest thanks go to Clarke Steinback for giving me the opportunity to direct for Chico State Game Studios during *High Noon*. This experience has changed my life, giving me a level of focus and dedication I had not thought possible. Of the many gifts you’ve given me over the years, this is the most precious.

Enormous thanks go to the *High Noon* Team, who are unfortunately too numerous to print here (check Appendix J for your name!), for your trust, support, and willingness to accept and try a new director’s nutty ideas. You not only helped me create an amazing Master’s project, you made Chico State Game Studios an amazing place to work. You’re a remarkable group and if I can find another team half as fun as you guys were, I’ll be the luckiest girl in the game industry.

Special thanks also go to the *D.A.V.I.S.* Creative Team: Andrew Holifield, Jeromy Maligie, Lindsey Anderson, Robert Utter, and Nathan Harling. Your dedication to Chico State Game Studios is inspirational and working with you on *D.A.V.I.S.* and *High Noon* was an honor.
I give further thanks to Andrew Holifield for providing me with balance and sanity through this crazy time in my life. I’m very lucky to have you, and you’ve done more for me than you can know.

Thank you to Nathan Harling for being one of my best friends and on-call editor. Your kind words of encouragement always helped me keep my momentum.

Big hugs and thank you’s go to Lindsey Anderson for being my Master’s Buddy. Without someone to do this with I fear I would have never finished. Thanks for being my friend, my rock, and my sounding board.

Most of all, I would like to thank my mother, Roxane Roberts, for her years of support and encouragement. Without you, I would not know the value of education, nor would I have the self-confidence to know that I could do anything I put my mind to. Any success in my life is truly a reflection of your wisdom and dedication to your children. I love you, Mom.

Thank you all for helping me get to this amazing point in my life,

~ Alisha Thayer
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publication Rights</td>
<td>iii</td>
</tr>
<tr>
<td>Acknowledgments</td>
<td>iv</td>
</tr>
<tr>
<td>List of Figures</td>
<td>viii</td>
</tr>
<tr>
<td>Abstract</td>
<td>ix</td>
</tr>
</tbody>
</table>

## CHAPTER

### I. Introduction
- Purpose of the Project | 2
- Scope of the Project | 7
- Significance of the Project | 7
- Limitations of the Project | 9
- Definition of Terms | 12
- Source Files | 12
- Chapter Summary | 12

### II. Review of Related Literature
- A Brief History of Video Games | 15
- Sources of Documentation | 18
- Methods of Production in the Game Industry | 19
- The Design and Implementation of *High Noon* | 24
- Chapter Summary | 26

### III. Methodology
- The Chico State Game Studios Team Structure: Senior Positions | 28
- The Chico State Game Studios Team Structure: Sub-Teams | 40
- The Production of *High Noon*: A Historical Recount | 56
- The *High Noon* Semester One Production Pipeline | 63
- *High Noon* Semester One Production Pipeline Observations | 76
<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Noon, Semester Two</td>
<td>81</td>
</tr>
<tr>
<td>High Noon Semester Two Production Pipeline Observations</td>
<td>90</td>
</tr>
<tr>
<td>High Noon, Semester Three</td>
<td>92</td>
</tr>
<tr>
<td>High Noon Semester Three Structural Changes</td>
<td>96</td>
</tr>
<tr>
<td>The High Noon Semester Three Production Pipeline</td>
<td>97</td>
</tr>
<tr>
<td>High Noon Semester Three Production Pipeline Observations</td>
<td>111</td>
</tr>
<tr>
<td>D.A.V.I.S., Semester One: Preproduction</td>
<td>114</td>
</tr>
<tr>
<td>Results</td>
<td>118</td>
</tr>
<tr>
<td>Chapter Summary</td>
<td>139</td>
</tr>
<tr>
<td>IV. Summary, Conclusions, and Recommendations</td>
<td>140</td>
</tr>
<tr>
<td>Changes Made to the Chico State Game Studios</td>
<td></td>
</tr>
<tr>
<td>Team Structure</td>
<td>140</td>
</tr>
<tr>
<td>Changes Made to Chico State Game Studios Policies</td>
<td>153</td>
</tr>
<tr>
<td>New Requirements of Chico State Game Studios Projects</td>
<td>167</td>
</tr>
<tr>
<td>New Chico State Game Studios Design Strategies</td>
<td>171</td>
</tr>
<tr>
<td>Cautions for New Directors</td>
<td>175</td>
</tr>
<tr>
<td>Conclusion</td>
<td>184</td>
</tr>
<tr>
<td>References</td>
<td>187</td>
</tr>
<tr>
<td>Appendices</td>
<td></td>
</tr>
<tr>
<td>A. Recommended Reading</td>
<td>190</td>
</tr>
<tr>
<td>B. Definition of Terms</td>
<td>193</td>
</tr>
<tr>
<td>C. High Noon Design Document</td>
<td>198</td>
</tr>
<tr>
<td>D. High Noon Syllabus</td>
<td>230</td>
</tr>
<tr>
<td>E. Example Assessment Exercises</td>
<td>232</td>
</tr>
<tr>
<td>F. Level Design Boot Camp Exam</td>
<td>241</td>
</tr>
<tr>
<td>G. Example Model Sheets</td>
<td>243</td>
</tr>
<tr>
<td>H. Modeling Requirements Checklist</td>
<td>246</td>
</tr>
<tr>
<td>I. Mid-Term/Final Self-Evaluation Example</td>
<td>248</td>
</tr>
<tr>
<td>J. The CSGS Team</td>
<td>250</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. The Chico State Game Studios Organizational Structure</td>
<td>28</td>
</tr>
<tr>
<td>2. The “Companion Cube” Reconstructed Church and the Final Reconstructed Church, Assembled by Level Designer Nathan Smart</td>
<td>48</td>
</tr>
<tr>
<td>3. The Proposed <em>High Noon</em> Pipeline for Semester One</td>
<td>65</td>
</tr>
<tr>
<td>4. Concept Design and Modeling Dependencies</td>
<td>77</td>
</tr>
<tr>
<td>5. An Example of a Level Paintover, painted by Concept Designer Jade Reed</td>
<td>83</td>
</tr>
<tr>
<td>6. Chico State Game Studios Inter-Team Dependencies</td>
<td>95</td>
</tr>
<tr>
<td>7. Enemy and Weapon Implementation During <em>High Noon’s</em> Third Semester</td>
<td>98</td>
</tr>
<tr>
<td>8. The <em>Unreal 2004</em> Red Shirt’s Revenge Ray Gun versus the <em>Unreal 3</em> Red Shirt’s Revenge Ray Gun</td>
<td>101</td>
</tr>
<tr>
<td>9. Production Cycles Illustrated with Chico State Game Studios’ Three-Semester System</td>
<td>119</td>
</tr>
<tr>
<td>10. Quality Assurance Checks for Game Assets</td>
<td>132</td>
</tr>
<tr>
<td>11. The Three-Semester System, Illustrated</td>
<td>155</td>
</tr>
</tbody>
</table>
ABSTRACT

HIGH NOON: DEVELOPING A VIDEO GAME PRODUCTION PIPELINE FOR CHICO STATE GAME STUDIOS

by

© Alisha Thayer 2009

Master of Science in Interdisciplinary Studies

Applied Computer Graphics

California State University, Chico

Summer 2009

Founded in 2004, Chico State Game Studios has produced four large-scale video games with team sizes ranging from eight to fifty-four students. The success of projects of this nature relies heavily on effective, documented production pipelines, something that past Chico State Game Studios projects have lacked due to the fact that no established knowledge base has ever been shared between incoming and exiting projects.

High Noon, Chico State Game Studios’ fourth project, sought to provide a venue for testing and documenting experimental production pipelines for use by large, student-run projects. This was done through extensive research, planning, customizing, executing, and documenting processes used through the author’s tenure as Director of
High Noon and as Art Director on the following Chico State Game Studios project, D.A.V.I.S.

The intention of this project is to present guidelines for incoming Chico State Game Studios teams by providing comprehensive documentation of the successes and failures of High Noon’s production. Moreover, this project stresses the importance of project documentation and aims to set a precedent for future Chico State Game Studios projects.
CHAPTER I

INTRODUCTION

Chico State Game Studios is an interdisciplinary, student-operated organization housed within the Applied Computer Graphics program at California State University Chico. Formally APCG 495: Advanced Production, Chico State Game Studios is a 3.0 unit course wherein students participate in the creation of a large-scale video game, from concept to final product. High Noon, the primary focus of this project, is Chico State Game Studios’ fourth game project and spanned three semesters of production. Created using Epic’s Unreal 2004 engine during its first two semesters and Epic’s Unreal 3 engine during its third semester, High Noon was a study in standardizing production pipelines for large-scale independent video game productions. As a member of Chico State Game Studios, I served as Director, authoring the game design document and back story, and leading a team of ultimately fifty-four students of diverse disciplines over High Noon’s three semesters of production.

Also explored during this project was preproduction on D.A.V.I.S., the successor to High Noon, which ran concurrent to High Noon’s third and final semester. As a member of the Preproduction Team and later as Art Director, I designed and implemented new organizational practices for both the preproduction and production based upon my experiences administering High Noon’s experimental pipelines.
High Noon was intended to provide Chico State Game Studios with an opportunity to experiment with new production standards and pipelines, resulting in the standardization and documentation of processes and best-practices for future Chico State Game Studios teams to use during production. Preproduction on D.A.V.I.S. provided a venue for testing new proposed preproduction standards based off of observations from High Noon’s informal preproduction.

This culminating activity is designed to not only demonstrate mastery of industry-standard practices and techniques, but to exhibit a synthesis of information and experience by researching, planning, customizing, executing, and documenting the preproduction and production of two large-scale electronic games. This thoroughly documented process is intended to be reproducible by future Chico State Game Studios teams.

This project is designed to serve as a resource for students participating in the Chico State Game Studios organization and independent game developers. Therefore, the reader is assumed to possess a working knowledge of computer graphics technologies, including: 3D modeling, animation, programming and scripting for games, and game design. Readers unfamiliar with computer graphics may encounter difficulty with discipline-specific methods and terminology.

Purpose of the Project

Chico State Game Studios was prototyped in 2004 in a small, independent study course through the Applied Computer Graphics Program. Formally established in 2005, Chico State Game Studios was designed to closely emulate game studios in
industry to give students an opportunity to contribute to large, interdisciplinary, collaborative projects similar to those that are produced in industry. Students who participate in Chico State Game Studios specialize in many diverse aspects of entertainment-based media: programming, 3D modeling and texturing, animation, game design, voice acting, art, music composition, video editing, web design, and print design.

Historically, projects at Chico State Game Studios have been selected annually from a pool of student-authored game ideas. Since its inception, Chico State Game Studios has produced four large-scale video games: Chicago Crux, Bronze Age, Viathan, and High Noon, all of which have been entirely conceptualized, managed, and created by students. D.A.V.I.S., Chico State Game Studios’ fifth game project, is currently in production, having recently completed its first formalized preproduction semester. The production life of the average Chico State Game Studios game is one academic year, with the exception of High Noon and D.A.V.I.S., which each span three semesters or one-and-a-half academic years.

Chicago Crux (2004) was the prototype project for Chico State Game Studios and an experiment in the feasibility of creating video games in an academically structured, student-managed environment. Chicago Crux was created using the Neverwinter Nights Aurora Toolset and possessed a small, eight-person production team of senior-level Applied Computer Graphics students. The subject of my undergraduate honors thesis in Applied Computer Graphics, Chicago Crux was my first venture into directing. While the project was never formally completed, it served well as a prototype class for the next, much more ambitious project the following year.
Bronze Age (2005) was Chico State Game Studios’ first game under the Chico State Game Studios label and first foray into the first person shooter (FPS) genre of gaming. Created using the Unreal 2004 engine, Bronze Age was a multiplayer competitive “shooter” set in the iconic Bronze Age of Greece. Instead of guns and rocket launchers, players used era-specific weaponry such as bows and swords, and fought other players in a “death match” style arena for points. An extremely ambitious project, Bronze Age boasted a customized “paper doll” system that allowed the player to customize the appearance of their avatar, as well as ten custom-designed and built levels.

For Viathan (2006), Chico State Game Studios chose to continue using the Unreal 2004 engine, in theory expanding the existing knowledge base and allowing for the creation of a more sophisticated game. Unlike Bronze Age, Viathan was a puzzle game. The player possessed only one “weapon,” which doubled as a tool for solving puzzles in the environment. Viathan was Chico State Game Studios’ first official experiment with custom artificial intelligence, single-player adventure games, and non-death match style gameplay. Gameplay in Viathan was largely linear, featuring seven custom levels, with project production peaking at thirty-three students (D. Schmittou, Chico State Game Studios, personal communication, March 2, 2009).

High Noon (2007 – 2008) was the first Chico State Game Studios project to remain in production for three semesters and the third Chico State Game Studios game to use the Unreal 2004 engine. Like Viathan, High Noon was a single-player, action adventure title with linear gameplay and custom artificial intelligence. Composed of four large levels and one complex boss fight at the endgame, High Noon was an exercise in complex storytelling in a fast-paced, 3D environment. High Noon was also the first and
only game in the Chico State Game Studios repertoire to incur a mid-production engine change, switching to a “next-gen” game engine in its third semester of production. *High Noon*’s experimental pipelines greatly contributed to its successor project, *D.A.V.I.S.*

*D.A.V.I.S.* (2009) is the first project to utilize the organizational structure, production pipelines, and game design theories developed and tested during work on *High Noon*. A psychological thriller, *D.A.V.I.S.* deviates from previous Chico State Game Studios productions in that it contains no weapons, no artificial intelligence, and only two characters: the player-character and the antagonist. Inspired by *Bioshock*’s (2K Games, 2007) focus on environmental storytelling, *D.A.V.I.S.* is an exercise in communicating a rich story to the player with relatively little interaction or other direct story delivery methods. Designed for the *Unreal 3* game engine, *D.A.V.I.S.* is Chico State Game Studios’ first true “next-gen” game, utilizing sophisticated computer graphics techniques and high quality standards.

There are many challenges with running an independent, student-operated game studio; for example, there is little precedent for how such an organization should be structured. *Chicago Crux, Bronze Age, and Viathan* all suffered from significant technological stalls and production complications due to a failure to document many of the processes and pipelines used during the projects. Additionally, documentation failure was one of the primary causes for losing many of the solutions discovered during each project for mitigating the significant *Unreal* engine learning curve. Although the manner in which Chico State Game Studios is operated is greatly inspired by professional game development studios, Chico State Game Studios is by nature a unique organization because of its student personnel and roots in academia. In essence, Chico State Game
Studios is run both as a class and as a game studio, integrating academic policies such as attendance and lectures with fast-paced, objective-driven game production.

Most modern video games are complex cinematic interactive experiences with sophisticated graphics, sound effects, music, voice acting, and gameplay elements. Video games are even beginning to rival other popular visual storytelling media such as television and film. Because of the inherent diversity in a large, interdisciplinary project such as a Chico State Game Studios game, it is imperative for a successful project to have clear, concise processes as well as a definition of that project’s production pipeline. With *High Noon*, I sought to mitigate common issues shared over *Chicago Crux*, *Bronze Age*, and *Viathan* by: formalizing the production pipelines used by Chico State Game Studios, creating standards for documentation, establishing a working knowledge base for the use of the *Unreal* engine to create games, and adapting these procedures to fit the unique academic environment in which Chico State Game Studios is housed.

It is my hope that future generations of Chico State Game Studios directors and staff use the pipelines, recommendations, and anecdotes presented in this paper as a foundation for their own projects, adapting and tailoring them as necessary; and that this project establishes a precedent for project documentation, in particular the director post-mortem (see Chapter IV, *New Requirements for Chico State Game Studios Projects, The Post-Mortem*) to ensure that the experiences of Chico State Game Studios teams are passed to their successors.
Scope of the Project

Semesters at Chico State Game Studios that were evaluated for this project include: *High Noon*, semester one (Fall 2007); *High Noon*, semester two (Spring 2008); *High Noon*, semester three (Fall 2008); and *D.A.V.I.S.*, semester one (Fall 2008). After each semester, the production pipelines that were used were evaluated and modified to better fit the subsequent semesters. This iterative approach to designing the production pipelines used by Chico State Game Studios allowed for testing many experimental techniques and tailoring standards specifically for use by Chico State Game Studios.

As this project is designed to give a foundation for game design and production as it applies to the Chico State Game Studios organization, the primary intended audience of this project is future Chico State Game Studios directors and team members. However, independent game developers, particularly those working in the *Unreal* engine, will certainly find the processes discussed in this project relevant.

Significance of the Project

Due to the competitive nature of video game production for profit, pipelines in industry are largely proprietary and are not freely shared with individuals outside of the company. In fact, for this project, I inquired with game companies such as Quantic Dream, Cyan Worlds, Bethesda Softworks, and others about their internal development practices and documentation standards. While there were a few smaller, independent studios that were willing to share their game design documents with me, larger studios closely guarded their production formulas. Each explained that it was against company policy to release information of that nature to the public.
Therefore, outside of textbooks, there are few documented resources for the aspiring game developer to utilize while learning to plan for the complex production of a game project. Student game developers must often work using trial and error and are generally inexperienced with the software and production techniques they employ. While trial and error has been known to work, it is generally because the average student game production possesses a relatively small team size, allowing for significant flexibility during production. As teams grow in number, so do the volume of linearly dependent tasks, and the ability to make swift changes to the pipeline significantly decreases. Additionally, in smaller teams many tasks are shared by the same people, making communication a considerably smaller obstacle compared to larger teams.

Designing projects for Chico State Game Studios is challenging given the differences not only from most game studios in industry, but from the average student development team. Unlike with most independent student gaming projects, Chico State Game Studios projects are particularly complex, generally requiring the production of a large amount of custom art assets and code. To achieve these high production requirements, team sizes on a Chico State Game Studios project reach around fifty, including external sub-teams such as writing and audio. Further, it cannot be forgotten that Chico State Game Studios is a classroom/game studio hybrid: the students working for Chico State Game Studios are not specialists, nor are they employees who can dedicate forty hours a week to the organization.

In general, this project is intended to provide a solid framework for large-scale independent video game development projects. The pipelines proposed during this project have been successfully tested for team sizes between eighteen and fifty-four, and
are tailored specifically for use with the *Unreal 2004* and *Unreal 3* engines. It is expected that readers of this project will use the outlined processes as a template for their own projects, building their own customized pipelines as needed. This project is designed to be an educational resource for aspiring game developers, detailing the processes from which games are made and the common mistakes made by novice designers.

More specifically, this project is designed to provide future Chico State Game Studios students with a frame of reference for their own productions. It is hoped that by considering this project in designs of their own, Directors will be able to mitigate common problems shared by all Chico State Game Studios projects thus far, in particular ill-conceived asset pipelines. The pipelines discussed in this project should inspire further development and refinement of the Chico State Game Studios production standards, increasing efficiency and ensuring retention of knowledge from project to project.

**Limitations of the Project**

Chico State Game Studios productions prior to *High Noon* were intended to be run as direct emulations of the game studios in industry. This led to inevitable complications; Chico State Game Studios is by nature both a class and a game studio. “Employees” at Chico State Game Studios are undergraduate students, most of whom have little knowledge of game production pipelines and practices due to limited course offerings in the Applied Computer Graphics program. In order to be effective game developers, students must be trained: taught to create model sheets, proper UV texturing techniques, modeling standards for video games, advanced texturing and surfacing techniques, et cetera, before they can produce high quality content for the game. Because
of this, Chico State Game Studios cannot be purely run like a game studio in industry; rather, it must function as a classroom first, providing students with the skills necessary to be successful contributors to the organization.

In fact, Chico State Game Studios can never fully operate as a game studio in its current state by nature of its implementation as an academic course at a California State University and high personnel turnover rate. Chico State Game Studios is a classroom first and a game studio second and therefore must create pipelines that consider the requirements of both entities. Because these pipelines had to be created on the fly as from old pipelines that focused only on the game studio aspect of the organization, *High Noon* did not run as smoothly or efficiently in its beginning semesters as it could have if pipelines had been established before the start of production with these factors in mind. Significant time was lost due to remaking assets, misunderstanding the pipeline, or having to research complicated game implementations.

Time is naturally an issue with any project, and was a significant factor when developing production standards during *High Noon*. Many experimental pipelines could not be fully tested in *High Noon*, and instead were left to the following project, *D.A.V.I.S.* Beyond preproduction, the results from these processes and their effect on *D.A.V.I.S.* are outside the scope of this project.

Personnel retention was a limiting factor in the beginning of *High Noon*, with most of the veteran team from *Viathan* choosing not to return to Chico State Game Studios for the *High Noon* project. This greatly contributed to a massive loss of knowledge; even though Chico State Game Studios had been working with the *Unreal 2004* engine for two years prior, little of that information was passed to the new project.
Rather than benefiting from the experiences of *Bronze Age* and *Viathan*, *High Noon* widely started with little to no existing knowledge of the *Unreal 2004* engine, or of lessons learned from the successes and failures of the previous two years of production. The *High Noon* project therefore required extra time to find its footing, with many tasks having to be re-implemented or even re-designed.

In addition, the lack of literature in this particular area poses a problem to students of game design who wish to pursue a complex electronic game project. Many elements of the game design process are not intuitive, and the process of creating a game is often too complex for a novice to learn with ease. The trial-and-error methods used in the creation of the *High Noon* pipelines were undoubtedly slow, leading to unused work, frequent bottlenecks, and mistakes in implementation that might have been lessened with existing research from which to base production models.

In Chapter II, the *Review of Literature*, this paper will discuss the history of game production, and explore the pipelines, literature, and web documentation researched for this project. Chapter III, *Methodology*, will provide an in-depth exploration of how Chico State Game Studios is operated, as well as the pipelines utilized during the three semesters of *High Noon*’s production and the preproduction semester of *D.A.V.I.S.* Problems and solutions encountered during production will also be discussed. Chapter IV, the *Summary, Conclusions, and Recommendations* provides the reader with an itemized list of relevant findings and explanations regarding standardized pipelines, newly implemented organizational structures, and *High Noon* and *D.A.V.I.S.*’s successes and failures.
Definition of Terms

For a detailed definition of terms relevant to this project, please refer to Appendix B.

Source Files

For additional information on this project, source files, or any other supplemental materials, please visit the author’s website at http://www.alishathayer.com.

Chapter Summary

In this chapter we have looked at the history of Chico State Game Studios and established that as a student-run organization it has unique requirements for the manner in which projects and teams are structured. Notably, Chico State Game Studios is responsible for not only an end product, but for educating teams on the techniques used to create professional quality games during development.

This project features the comprehensive documentation of the life of a large-scale project and is to be used as a source of information and structure for future Chico State Game Studios Directors and independent game developers. By doing so, this project aims to provide a precedent for the documentation of game projects at Chico State Game Studios, and to increase the information available to aspiring game producers. Because of the technological nature of video game development, a working knowledge of computer graphics terminology and processes is expected of the reader.
CHAPTER II

REVIEW OF RELATED LITERATURE

In order to understand the nature of video game production today, it is important to recognize the modest origins of the industry and the accelerated rate at which the industry grew, as well as current methods used in game production. Historically, changes to production pipelines often occurred as demands of games grew with advancements in hardware and computing power. Moreover, there currently exist no publicized, standardized practices for the development of video games. Though popular software engineering methods are typically looked to for models of video game creation, most professional companies use a proprietary method, hybridizing software development practices and unique game project demands.

The video game industry was largely born from the garages of technology enthusiasts in the 1960s. For years, video games were created by small teams of industrious programmers with no need for production standards, documentation, or even artists. The creation of a video game was a highly technical process, often involving the need for custom hardware and clever engineering. “In the beginning, programmers were responsible for creating every element of their games. The same person who created the concept was also responsible for the programming, art, and even sound effects” (Kent, 2001, p. 182).
Naturally, the need for standardized team structures and production pipelines did not exist in the early days of games; methods of communication and documentation are overbearing and nonsensical when a production team is composed of two or three people. However, as team sizes and game complexity grew, so too did the need to manage produced assets and define project tasks. Because the industry grew so quickly, most of these processes were designed out of necessity while games were in production, resulting in a massive lack of cohesion in practices used by industry and an equally massive lack of documentation. Detailed project documentation released to the general public is few and far between.

Video games today are worlds different from their 1970’s counterparts. Composed of complicated physics systems, true-to-life high-resolution textures, high-polygon 3D models, complex animation systems, sophisticated artificial intelligence, and multi-channel sound, video games can be complex cinematic experiences that take years to make with production teams exceeding a hundred strong. Moreover, the manner in which games are designed is significantly different than in the days of *Pac-Man* and *Space Invaders*. No longer are games designed to be quarter-eating parlor entertainment, but instead are designed to entertain and engage players in their homes, much like movies and television.

Dan Irish, producer of *Myst III: Exile*, described the current state of the industry as follows:

The game industry is still young. Founded just three decades ago, the evolution of the video game industry continues today, while the race to maturity is still far from over. The breadth of the appeal is constantly growing with each new game. By exploring ways to expand as well as to take compelling experiences to new depths, we get one step closer to that maturity. (Irish, 2005, p. xxii)
Indeed, the footprint of the industry is growing at a rapid pace, and game creation has become accessible to anyone with a home computer and basic knowledge of computer graphics and programming. Not only are games made in professional studios, but large followings have developed for “indie,” or independent games. Independent games like *Braid* (2008) and *World of Goo* (2008), developed with incredibly small teams, have been acclaimed for their innovation and refreshing contribution to the industry by providing fresh new ideas for gameplay and game subject matter.

Students aspiring to be game designers in today’s market are faced with choosing between developing independently or becoming specialists in an aspect of game development and working for a large game developer. Either path can prove to be a difficult one; independent games are largely made by hobbyists or one-to-two person teams and do not guarantee monetary compensation upon completion. Studios, however, rarely hire staff with no industry experience. Students wishing for careers in the game industry have very few documented resources to rely on for learning about production and required skills, with most information coming from websites such as *Gamasutra* or the monthly publication, *Game Developer*.

A Brief History of Video Games

Video games received their first spark of life in the late 1950s when physicist Willy Higinbotham created a simple game, *Tennis for Two*, on an oscilloscope. A couple of years later, *Spacewar*, the first interactive computer game was created by Steve Russell at MIT in 1961. *Spacewar* was a simple, vector-based game, and allowed two
players to shoot at each other from small, geometric spaceships. *Spacewar* remains popular today, and from it has derived many similar games during its thirty-year lifespan.

The definition of the title “Father of Video Games” is a controversial one, due to the fact that the first coin-operated arcade game and the first home console were shipped within six months of one another (Kent, 2001). Nolan Bushnell, founder of Atari and developer of *Computer Space*, a game inspired by *Spacewar*, is considered to be the father of the video game industry because he was first to sell video game units for profit. Ralph Baer, the driving force behind the Magnavox Odyssey (released in 1972) and the designer of *Ping-Pong*, is considered to be the father of home gaming consoles (Hawkins, 2006). These two events marked the founding of the game industry and led to a massive boom in computer and arcade gaming alike, laying the foundation for gaming as it is today.

Gaming boomed so quickly, in fact, that in 1978 the popularity of *Space Invaders* caused a national coin shortage in Japan, forcing the Japanese mint to triple the production of ¥100 coins (Kent, 2001). Culture formed around the playing and developing of video games, producing magazines, special interest groups, television shows, and later websites and blogs dedicated solely to tracking advancements in the game industry.

The video game industry was dominated by arcade and console gaming up until the early 1990s when the popularity of personal computers allowed for the development of complex games on PC and Macintosh computers. Though games were being developed for computers prior to the 1990s, they were generally not accessible to wide audiences because of the high cost of personal computing. Between the years of
1972 and 1986, a plethora of home gaming systems were released from companies such as Magnavox, Fairchild Camera, Mattel, and Atari.

The development of sophisticated home console systems resulted in the first major console war in 1986 with the worldwide release of the Nintendo Entertainment System (NES), the Sega Master System, and the Atari 5200 (Kent, 2001), establishing a precedent for generational releases of major home consoles. With the release of the NES, games began to show potential as a story-telling medium, resulting in such games as *The Legend of Zelda* and *Final Fantasy*. These games contained characters, dialog, and varying objectives; a significant departure from the arcade games such as *Computer Space* and *Donkey Kong* on which the industry was founded.

The shift from games as a competition for high scores and the rising focus of games as an entertainment medium meant the development of longer, more complicated titles for home consoles and PCs alike. The 1990s marked the growing popularity of the adventure game, a puzzle-driven genre that sought to create detailed game worlds and stimulating stories. Perhaps the most successful game of this genre was Cyan World’s *Myst*, released in 1993 and selling more than twelve million copies worldwide with its sequel, *Riven* (Cyan Worlds, 2009). Also in 1993, id Software produced *Doom* for PC, the first major first-person shooter, which solidified the first-person shooter genre and spurred hundreds of similar titles in later years.

The years following the release of *Doom* brought games that continued to push the computing capabilities of the hardware for which they were developed. The popularity of games continued to grow, spurring multiple iterations of console wars between the big-name players in the industry: Sony, Nintendo, and Sega, who was later
replaced with Microsoft after the death of the Sega Dreamcast and the release of the Microsoft Xbox. PC gaming continued to thrive, surpassing the capabilities of the home consoles for a time.

The current home consoles (Sony Playstation 3, Microsoft Xbox 360, and Nintendo Wii) are as powerful as a moderately expensive home PC, with multiple graphics processors, gigabytes of internal storage (the Wii being an exception with only 512 megabytes of storage), and high-speed networking capabilities. Handheld gaming has also found a significant niche in cellular telephones, iPods, and handheld consoles like the Nintendo DS and Sony PSP. Gaming on multiple platforms (PC, console, and portable) is more popular now than it has ever been, with consoles and games selling at impressive rates. In 2009, Nintendo announced that their Nintendo DS had sold more than one-hundred million units worldwide in the five years since its release, approaching the Nintendo Game Boy’s record of one-hundred and sixteen million units sold over sixteen years. The current system with the most units sold is the Sony Playstation 2, which has sold over one-hundred and fifty million units since its release in 2000. (Watts, 2009).

Sources of Documentation

Though the documentation of game development is only just beginning to occur in industry, there are a significant number of valuable resources available to prospective game developers that were used during the course of the High Noon project. Foremost of these resources is the massive archive of industry-authored articles, post mortems, job postings, and career guides housed at Gamasutra.com.
Gamasutra is owned and operated by Think Services, who also publishes Game Developer Magazine, a monthly print publication on advancements in game development technology and processes. Notably, Game Developer Magazine publishes a monthly post-mortem, ranging from AAA titles such as Little Big Planet to smaller titles like The Maw. Books are naturally valuable resources as well, with noteworthy titles including Dan Irish’s The Game Producer’s Handbook and Tracy Fullerton’s Game Design Workshop.

Because of weak documentation on former Chico State Game Studios projects, High Noon was unable to draw upon the design documents or other observations of the projects that came before it. Instead of designing a high-level game document based off of Chico State Game Studios precedent, High Noon’s design document was created using examples from generous game developers who were willing to donate their documentation to the project from completed projects. These unpublished documents were distributed via e-mail and were intended for academic purposes only.

Methods of Production in the Game Industry

In his 2005 book, The Game Producer’s Handbook, author and game producer Dan Irish explains four software production methods used in the game industry. While some methods are very simple and produce many risks for a project, many of these methods are derived from practices used in software engineering, tailored to fit the needs of a game production.

Code Like Hell, Fix Like Hell or Extreme Game Development

Irish explains the “code-like-hell, fix-like-hell” method of game development as one of the oldest and most common models used to create games. This is unsurprising
considering the history of gaming, with many “studios” starting as experimental projects in hardware hacking, and many successes being happy accidents or cases of “right place, right time.”

Using the “code-like-hell, fix-like-hell” approach, programmers create code as fast as possible according to what they believe the design of the game calls for. Produced code is then tested and fixed. This process occurs iteratively until the game is completed. Irish points out that this method is known to leave errors and bugs undiscovered for long periods of time, risking newer code having similar flaws and muddying production. “This model is generally only suited for small projects with simple requirements,” Irish explains, “because the code is difficult to maintain over a longer period (six or more months)” (Irish, 2005, p. 16).

**Increments to Completion**

By using the increments-to-completion method, projects are broken up into small, modular elements that are designed to be completed independently of the rest of the project. All of the modular elements are divided up by teams, and then compared against the high-level design document. At the beginning of the project, specifications for game elements are typically only designed at the high-level, leaving the low-level feature design to a consensus between the designers and programmers. Documentation, therefore, does not occur until just before or just after the feature has been implemented.

This method is flexible because it allows for many parts of the project to be developed in parallel, but it can cause serious issues if not managed properly. In addition, the increments-to-completion method commonly results in a significant amount of
revised or redesigned work. However, this method lends well to the application of lessons learned during the life of the project.

**Iterate Until You Drop**

As the name implies, the iterate-until-you-drop method involves revisiting implemented elements in an iterative fashion, adding additional improvements with each iterative pass. This method is highly flexible, allowing for significant midstream changes of the game’s design and feature list. In addition, this method’s tremendous flexibility allows for quick adaptation to changing market demands.

However, Irish warns that this method can get out of hand quickly if the producer implementing it does not possess a strong understanding of object-oriented programming (OOP) and software development. Deadlines can get out of control, and the team can find themselves on a “never ending treadmill” of software revisions that can quickly balloon a project’s timeline and budget.

**Agile Development**

Of all of the methods presented by Dan Irish in *The Game Producer’s Handbook*, Agile Software Development seems to be the most widely supported and accepted in the game industry today. “Agile project management [is] an excellent process for organized and disciplined teams,” says Irish. Agile development has five key stages: Envision, Speculate, Explore, Adapt, and Finalize.

In the Envision stage, the project manager asks a number of questions:

- What is the game?
- What makes the game fun?
- What is the scope?
• What is the audience?
• How will the team work together?

The goal of the Envision stage is to gain an understanding of the game project and to start spreading enthusiasm to the team members who will be working on the project. The initial vision of the project is established, as well as the direction that project lead sees is appropriate for implementing of the game.

During the Speculate stage, Irish explains that the game’s high-level requirements are determined, along with an outline of the work that is to be completed for the project, a development plan, a tentative schedule with resource allocations, a list of features, detailed risk management plans, and a budget.

The team uses the Explore phase to identify features as defined by the game’s high-level design document and implement them. The features outlined in the design document are first priority, and then further development of features is achieved by the formation of a collaborative project community on the team. This team is largely self-managed and organized, running parallel to the project lead, leaving the lead to explore more administrative tasks.

The Adapt stage is vital in the success of an agile project. Projects must adapt to schedule changes as necessary rather than trying to make the project conform to an unsuccessful schedule. More importantly, the Adapt stage allows for the incorporation of lessons learned during the project midstream, rather than having to wait until the end of the project’s life to complete an analysis of the project’s successes and failures. The project lead looks carefully at the game in its current stage, checking for its fidelity to the
original game plan, and makes changes to the team, pipelines, schedule, or game as necessary.

The Finalize stage, like the name implies, is the stage in which the project is completed. But not only that; the Finalize stage includes the documentation of all of the successes, failures, and discoveries on the project to ensure that future projects learn from the experiences of the completed one.

Agile project development seems well suited for game development, which often has many hidden requirements or technological hurdles to overcome before the game can be completed. Indeed, few game projects are implemented with one-hundred percent faithfulness to the original high-level design document. Agile project development is designed with change in mind, but more importantly with the intention of improving future projects and learning from mistakes, rather than repeating them.

Top-Down Planning versus Bottom-Up Planning

Planning for game development projects generally occurs in one of two ways: Top-Down or Bottom-Up. The Top-Down approach to project planning generally involves a single person designing the inclusive plan for the project’s design and implementation. This schedule is usually more like a projection of what production might be like, and can be frustrating for team members because it removes control from most people working on the project. Instead of being a part of planning their own tasks, team members instead receive instructions from “on high,” which can at times seem disconnected or extraneous.
Because the schedule is simply a projection, it runs the risk of being adhered to too closely. “At best, it’s a guess; at worse, it’s totally wrong,” (Irish, 2005, p. 22). Irish warns that schedules designed using a Top-Down approach should be closely guarded and released wisely to the rest of the team, and always with a reminder that changes are expected to occur.

The Bottom-Up approach features the involvement of relevant team members in the design of the production schedule. This method is generally more accurate because of the diverse technological and experiential backgrounds of the team members involved in planning. Together, this planning committee identifies all of the game’s planned art assets, game features, and other important project requirements. Irish suggests that some preproduction occur before the committee begins to design the project’s schedules to facilitate relevant and appropriate decisions.

The foremost goal of the committee is to determine short-term goals, and then apply a clear schedule to their planned completion. Irish cites the importance of involving all people from all disciplines in the development of the game, placing emphasis on personal ownership and pride.

The Design and Implementation of

*High Noon*

*High Noon* employed the “code-like-hell, fix-like-hell” approach during the first semester of production by default due to a lack of familiarity and understanding of software development methods. Like Irish warns, a significant portion of work developed during the first semester was lost or simply had to be redone because it was constructed with faulty information or incomplete knowledge of the development tools. Ultimately,
the “code-like-hell, fix-like-hell” approach was not a good fit for the *High Noon* project, and other methods of production were pursued in later semesters.

As the production methods for *High Noon* evolved, they began to take on characteristics of Agile project development. Though *High Noon* did not explicitly work in stages, the concepts explored during each of the major Agile stages were emphasized more and more as *High Noon* progressed through production. Adaptability, learning, and documentation became foremost goals for the project, and were at the core of the design of *High Noon*’s successor project, *D.A.V.I.S.*

The “code-like-hell, fix-like-hell” approach was a natural choice for a novice team of untrained game producers, and it was not until the Chico State Game Studios team developed a reliable core of dedicated, trained, disciplined, and knowledgeable team members that the production methods used to create *High Noon* began to gravitate towards Agile methods.

*High Noon* was designed using a top-down method, and encountered many of the issues that Irish raises in *The Game Producer’s Handbook*. Because the specifications for *High Noon* were designed by only the Director, many faults existed with game’s design and scheduling that were not discovered until late in production. With the start of *D.A.V.I.S.*, Chico State Game Studios experimented with using a bottom-up approach by implementing a Preproduction team consisting of veteran Chico State Game Studios team members with diverse technical backgrounds. The results of this change were staggeringly positive; designs generated for *D.A.V.I.S.* were more robust than *High Noon*’s ever were, and the technical requirements of *D.A.V.I.S.* were explored much earlier in the project by qualified team members.
Chapter III will explore the working Chico State Game Studios structure, as well as an in-depth look into the growth of *High Noon* over its three semesters of production, detailing its successes and failures as well as modifications to the pipelines used during the duration of the project. By breaking down each semester by team responsibilities, Chapter III endeavors to provide important insight into the pipeline development experiments that occurred during *High Noon*’s production.

**Chapter Summary**

By understanding the origins of the game industry and some of the current implementation methods used in game production, this chapter aims to provide a foundation for understanding the techniques used during the production of *High Noon*. Exploring the beginnings of the video game industry provides us with a first glimpse at what would quickly evolve into the production planning juggernauts of today, and provides context for why production methods in industry are structured as they are currently.

Furthermore, understanding the methods that were used during this project provides a frame of reference for the narrative recounts of *High Noon*’s production presented in subsequent chapters. The parallels between the common issues in industry-level game production and the issues raised by *High Noon* for Chico State Game Studios highlight the similarities and differences between Chico State Game Studios and the game industry today.
CHAPTER III

METHODOLOGY

Chico State Game Studios is currently structured according to the experimental pipelines designed during the three semesters of production on *High Noon* and the preproduction semester of *D.A.V.I.S*. This chapter will outline in detail the duties of every participant of the Chico State Game Studios team, the global duties of each team, the production pipeline for making a large-scale video game using *Unreal* technology, and a detailed outline of recommended practices for each semester of the Three-Semester System (see Chapter IV, *Changes Made to Chico State Game Studios Policies, Introduction of the Three-Semester System*).

This chapter also provides a detailed recount of the iteratively designed pipelines used during the three semesters of *High Noon*’s production. Each semester of production will be looked at individually, highlighting the pipelines and team organizations used and how they evolved from the results of the preceding semesters.

It should be noted that production processes are malleable and should be tailored to suit the individual needs of each project as necessary. This chapter is intended for the novice Director who may not be intimately familiar with the requirements of creating a large-scale video game with over fifty people actively working on the team. Video game production can be a Herculean undertaking, requiring an experienced, flexible, interdisciplinary team in order to bring an idea from concept to fruition.
This chapter is intended to impart to the reader an intimate understanding of the complex processes and inter-team relationships necessary in creating a sophisticated, “next-gen” video game. New Chico State Game Studios Directors should use this chapter as a planning tool: noting the pipeline and outlined dependencies, the roles of the different team members, and the recommended semester activities and integrating them into their plans for their own Chico State Game Studios productions. See Figure 1 for a graphical representation of the Chico State Game Studios team structure.

![Figure 1. The Chico State Game Studios organizational structure.](image)

The Chico State Game Studios Team Structure: Senior Positions

**Executive Producer**

The Chico State Game Studios Executive Producer is the Applied Computer Graphics faculty member responsible for the oversight of APCG 495: Advanced Production. He or she maintains close communications with the student Director throughout the life of the project and is the final quality check on content generated for the game. The Executive Producer also provides final resolution on any team conflicts or issues that may arise in class. At the end of a production cycle, or semester, it is the Executive Producer who assigns final grades.
Usually serving as Chair of the Chico State Game Studios Executive Committee, the Executive Producer organizes monthly meetings, sets agendas, and ensures all issues put forth by the Director are addressed in a timely manner.

**Director**

The Director of a given Chico State Game Studios project is usually an undergraduate student who has worked with Chico State Game Studios long enough to be comfortable with the technology and procedures used in making large-scale video games. A student with senior standing, the Director’s core function to the team is to provide a structural backbone to the project and ensure that all sub-teams are actively working toward the same goal. The Director’s position on the Chico State Game Studios team is that of support and is primarily administrative, though he or she will produce content on occasion if the team lacks the staff for a particular job. However, the Director is largely not a content producer, and when the pipeline is running smoothly, the Director never physically touches the assets made for the game.

The Director’s organizational duties begin almost immediately after he or she agrees to direct a Chico State Game Studios project. Before the game even goes into preproduction, it is the duty of the Director to create a plan for each phase of the project, marking important deadlines and milestones and ensuring that all collaborators are aware of the director’s vision. During preproduction, the Director authors the original game design document (see Appendix C) with the Preproduction team, and leads Preproduction team meetings. Additionally, the Director is responsible for writing and delivering public speeches about the game’s progress.
At the project’s start, the Director organizes the students in the Chico State Game Studios class into sub-teams based off of their interests and existing skill sets. It is the duty of the Director to verify that there are enough students on each sub-team to meet the requirements of the game. In the event that any team is short, the Director then scouts talent by soliciting various academic departments (English, Music, Computer Science, et cetera) for additional students. The Director is the primary liaison between Chico State Game Studios and the rest of the University, using resources like the organization’s webpage and the game design document to attract new talent to the project.

In the classroom, the Director runs day-to-day operations: authoring the class syllabus, taking role, administering Mid-Term and Final Self-Evaluations, and resolving any team schedule and/or personality conflicts that arise over the course of the project. During class, the Director is responsible for ensuring that all teams are aware of their current tasks and the attached deadlines, as well as ensuring that all teams have the resources that they need to function. In this capacity, the Director largely functions as a matchmaker for the project, connecting experts to other experts to keep the flow of resources and information running smoothly.

The Director is one of the most important creative problem solvers on the project, as he/she must closely monitor the progress of both the team and the project at all times. He or she actively checks for lulls in productivity, decreasing morale, or inconsistencies in the assets created for the project. It is also one of the Director’s foremost duties to identify errors in the production pipeline and to quickly mitigate them by making alterations to production standards and/or the pipeline itself. In the classroom, the Director typically does not produce work, but rather moves purposefully from sub-
team to sub-team to check in, monitor work, and make certain that all work is adhering to the established standards for the game. The preponderance of the Director’s planning and documentation work, however, is done outside of the classroom.

Technical Directors, Research Leads, the Assistant Director and the Art Director all report regularly to the Director on the progress of their sub-teams. This line of communication assists the Director with making accurate projections for milestones and major deadlines. Further, the Director is at the mercy of his or her management staff, who provide on-call assistance for any issues that may arise out of the project. In addition, the Director and the team leads all meet once weekly to discuss game and staff issues as a group, ensuring that all sub-teams are aware of one another’s progress and needs, and ensuring that work is delegated appropriately.

The Director is the final check in quality control of assets and code that are integrated in to the game; regularly checking to see that all work completed for the game adheres to the vision and direction of the project. Similarly, the Director determines when the game is ready for regular playtesting, schedules playtest dates, and carefully defines the goals and projected outcomes of each playtest.

Outside of the classroom, the Director is also a member of the Chico State Game Studios Executive Committee by default, and meets with them once a month to discuss the project’s progress and discuss any production issues. Additionally, the Director provides a direct line of communication with the Executive Producer of the project, reporting progress and plans for following weeks. This level of oversight ensures that the Director, while independent, does not lack solid support for difficult organizational and/or staff issues that may arise.
Assistant Director

The Assistant Director works closest to the Director out of any member of the Chico State Game Studios team. Where the Director is tasked with the generation of ideas, procedures, and methods for completing the game project, it is the Assistant Director’s role to see that they are implemented and understood by the team as a whole. The Assistant Director’s function is nearly purely administrative, focusing on asset management, documentation, and the dissemination of knowledge.

The Assistant Director is a senior-level student who has worked for Chico State Game Studios for long enough to be comfortable with the technology and production pipelines used by the organization, has intimate knowledge of the processes related to game design and creation, and can closely monitor the flow of assets and the effectiveness of pipelines throughout production. Like the Director, the Assistant Director is a valuable problem solver, responsible for quickly identifying problems and developing inventive solutions on the fly. If procedures are not working or need to be modified to better suit the project, it is the responsibility of the Assistant Director to quickly alert the Director and find resolution. Should the Director become unavailable for any reason, the Assistant Director serves in his or her absence.

Organization and distribution of information is the Assistant Director’s foremost duty, ensuring that all team members are aware of the current status of the project. They iteratively publicize to the team the number and quality of assets created at any given time and the tasks that are yet to be completed. The Assistant Director creates and maintains detailed spreadsheets of assets, milestones, and assignments and publishes them regularly to the class. When necessary, the Assistant Director supports the Director
and Technical Directors by supplying the team with feedback on assets and game implementation progress.

Assets are generated at such an accelerated rate on a Chico State Game Studios project that there must be a dedicated team member to track them as they come in, check them for naming conventions and functionality, and to add them to a working database. The Assistant Director is well suited for this job because of their inherent knowledge of the project and the expectations of the Director. Without a dedicated team member on asset management, there is a high risk of losing work, creating duplicate assets, or creating assets that do not fit the desired specifications for the game.

Like the Director, the Assistant Director serves as a surrogate team member for any teams that are lacking expertise and/or manpower to complete the tasks required of them. Because of his or her senior class standing, the Assistant Director should have a working knowledge of all of the disciplines involved in game creation. Therefore, it should be easy for the Assistant Director to float from sub-team to sub-team, providing informational expertise or directing external experts to the sub-team as needed. Because the Assistant Director closely monitors the progress of all of the sub-teams, it should be immediately apparent to him or her when sub-teams are lacking resources or structure.

It is advised that the Assistant Director be familiar with server management and version control software, such as Apache, Trac, and Subversion. Because of the Assistant Director’s intimate knowledge of the production pipelines used for the project, as well as the flow of assets through those pipelines, allocating server and version control software management to the Assistant Director is a natural fit. Though this job can be assigned to an external expert with server administration experience, to be able to give
this responsibility to the Assistant Director is highly beneficial to the team because of his or her familiarity with the game and its production requirements.

Above all things, it is the Assistant Director’s duty to ensure that he/she and the Director work as a tight-knit team; always closely corresponding to verify that consistent information is being disseminated through the team and that problems are addressed quickly and efficiently. Along with the Director and Technical Directors, the Assistant Director provides an important organizational and structural backbone amidst the chaos of production.

**Technical Director**

All of the major sub-teams on a Chico State Game Studios project (except for Writing and the Preproduction team) have one assigned Technical Director (see Figure 4.1). A Technical Director is a resident expert, proficient in the technology used by their respective sub-team, and works closely with the sub-team members on creating production quality content for the game. Technical Directors closely monitor their sub-team: regularly distributing assignments, running active quality assurance, troubleshooting, mentoring and tutoring, and providing critique on submitted game assets. Each sub-team works as a unit under the Chico State Game Studios umbrella, and for all intents and purposes, the Technical Director is to the sub-team as the Director is to the whole team.

The Technical Director works closely with the Director and Assistant Director to ensure that deadlines passed on to their sub-teams are consistent with the Director’s established milestones. Like the Director and Assistant Director, the Technical Director also monitors the working production pipelines, checking for bottlenecks or necessary
changes to better fit the project, or their sub-team in particular. They provide advisory support to the Director and Assistant Director for developing new standards for production, constantly seeking ways to adjust the pipeline and increase the number of completed assets flowing into the game. In particular, the Technical Director ensures that their team members closely adhere to the production standards and guidelines developed by the Assistant Director and Director, including naming conventions, efficiency standards (such as polygon count or code efficiency), and asset submission procedures. The Technical Director also provides frequent updates on team progress to the Director and Assistant Director.

In addition to monitoring pipeline-related elements like milestones and bottlenecks, the Technical Director also assists the Research Lead in creating new production standards for non-native game requirements, or game elements that do not already exist in the *Unreal* engine and must be custom designed and engineered. With the Research Lead, the Technical Director commonly works cross-team, drawing off of the knowledge and experience of other Technical Directors and team members to quickly find solutions for complicated design problems.

Outside of administrative duties, the Technical Director also produces content for the game. For example, because of their existing mastery of the pipeline and involved technologies, Modeling Technical Directors are often tasked with producing high-priority, complex assets to specifications.

**Research Lead**

The Research Lead position was introduced to the Chico State Game Studios team structure in response to the transition to the sophisticated *Unreal 3* engine. The
switch to *Unreal 3* created a demand for the development of research and standards for many “next-generation” game making technologies, such as advanced surfacing techniques, visual scripting, and higher resolution meshes. During production on *High Noon*, the knowledge base of best practices in *Unreal 3* established by the Research Lead was integral to the game’s success in its third semester, and in the success of *D.A.V.I.S.*, the project following *High Noon*. Even as Chico State Game Studios’ familiarity with the *Unreal 3* engine grows, the Research Lead remains central to the functionality of any sub-team, and provides a number of other managerial services outside of research on new technologies.

Essentially, the Research Lead is a team member dedicated to finding information on the technologies essential to the game’s implementation, and searching for solutions to problems as they arise. When not actively researching, the Research Lead acts as a hybrid Technical Director and content producer, and therefore carries many of the Technical Director’s responsibilities as needed, especially if the demands of a sub-team are too strenuous for a single Technical Director.

Like the Technical Director, the Research Lead keeps a watchful eye on the effectiveness of the production pipeline and its effect on both the flow of assets and the productivity of the team. Because of their knowledge of the *Unreal* engine’s inner workings, the Research Lead is a valuable resource for the Director in predicting the complexity of game implementation tasks, and can greatly assist the Director in more accurate projections for milestones and deadlines.

The Research Lead’s primary duty is to document the knowledge they acquire and to publish it for the rest of the team, and for future Chico State Game Studios
projects. Without the Research Lead actively dedicated to documentation, it is likely that
documentation gets pushed to the end of the schedule, ultimately forgotten about when
the project reaches a close. While this may not affect the current project, subsequent
projects suffer as they cannot learn from the experiences of the project prior. In the
absence of technologies to research, the Research Lead should at dedicate themselves to
the documentation of information for future Chico State Game Studios teams.

Art Director

The Art Director is, surprisingly, a new position in the Chico State Game
Studios organization, introduced during the preproduction phase of D.A.V.I.S. to mitigate
issues surrounding inconsistent visuals and a lack of enforced quality standards
experienced during High Noon.

The Art Director works as a member of the Art Assets team and directly
manages the Concept Design team as the Technical Directors do with the rest of the
major Chico State Game Studios sub-teams. It is important that the Art Director have a
sound understanding of the flow of assets through the organization, as their decisions
regarding the order in which items are concepted and generated ultimately affects the
order in which the game is implemented. Ordering concepts vital to the game’s
functionality late in a production cycle can have severe and adverse affects on a game’s
production and cause massive bottlenecking later in the production.

It is the duty of the Art Director to interpret the Director’s specifications for
time period, color palette, mood, setting, et cetera, and from them create a distinct visual
style that can be enforced throughout production. This visual style is then communicated
to the Concept Design team, who uses it as a framework for the assets (such as model
sheets and cinematic art) that they create during the project. The Art Director also works closely with the Programming team to create user interface (UI) elements such as in-game menus, ensuring their consistency with the game’s visual standards.

The Art Director provides major quality assurance checks on all of teams responsible for generating art assets or otherwise effecting the visual interpretation of the game: Technical Art, Modeling, Concept Design, and Level Design. He or she runs active quality assurance on all textures generated by the Modeling team and model sheets generated by the Concept Design team, checking for style consistency and fidelity to the game world. Next to the Director, they are the foremost authority on the visual interpretation of the game. By providing close oversight on all art-generating teams, the Art Director ensures the assets generated are unified and that they fulfill their purpose to the game: to tell a story and immerse the player into the game world.

Ideally, the Art Director’s experience with previous Chico State Game Studios projects will lend well to their understanding of art generation and texturing for video game productions, making them a perfect candidate for a Texturing Advisor on the team. Currently, the Applied Computer Graphics program does not offer any courses on texturing for video games, widely resulting in a deficiency of experienced texture artists in the organization. With D.A.V.I.S., the Art Director was responsible for offering specialized lectures on best practices for using Adobe Photoshop for Unreal 3, as well as proper typography use and simple design principles. These lectures were necessary in preparing the art teams for creating a unified product, ensuring all team members were using the same set of tools in the generation of their concepts and textures.
Like the other leads on the Chico State Game Studios team, the Art Director is a content producer when needed. As they are often an experienced generator of content for video game projects, it is common that the Art Director will create or participate in the creation of high-priority or frequently seen game elements, promotional materials, web site designs, and UI designs.

Technical Art Lead

The Technical Art Lead is responsible for maintaining quality and consistency among assets as they are imported into the game. With the Technical Art team and the Art Director, the Technical Art Lead defines the final aesthetic properties of the 3D assets, particles, and specialized surface materials for the game. For example, if it is projected that the game will use a lot of chrome and neon, the Technical Art Lead will set up material creation methods for chrome and neon in *Unreal 3* that suit the overall look and feel of the game. In addition, the Technical Art Lead and the Level Design team are closely affiliated, as it is the Technical Art team’s responsibility to ensure that assets are working properly in-game.

Because technical art marks the end of an asset’s journey through the production pipeline, it is crucial that during the importation process assets undergo their last quality checks: verifying artistic consistency, technical accuracy, and naming conventions. In addition to the Technical Art Lead, the Assistant Director and Art Director lend their eyes to making sure that all assets exiting the pipeline are game-ready and error free.
The Chico State Game Studios Team
Structure: Sub-Teams

Concept Design Team

The Concept Design team marks the beginning of the production pipeline for most visual assets in the game. Though the Director and the Preproduction team are tasked with writing the game in terms of plot, characters, game play, and objectives, the Concept Design team’s job is to provide all necessary art and visual planning for the game. The Concept Design team is one of the only Chico State Game Studios teams that actively works with all of the major sub-teams, providing concepts, artwork, and designs to Modeling, Animation, Level Design, and Programming on a regular basis. Decisions made by the Concept Design team ripple through the entire production pipeline, serving as a creative foundation for the rest of the Chico State Game Studios team.

Concept Designers are the only Chico State Game Studios members who are explicitly responsible for researching the period and visual style of the game, ensuring that all visual assets are consistent with the Director’s intentions. For example, during the production of *D.A.V.I.S.*, Concept Designers were tasked with researching the visual style and architecture of the 1950s, becoming experts on even minute details such as the style of kitchen sinks popular during the era.

As resident experts on the game’s visual style, one of Concept Design’s primary functions during the beginning of a game project’s life is to create a constant flow of model sheets to pass to the Modeling team for the creation of all in-game 3D assets. Model sheets contain vital information for the creation of any custom asset, from small assets such as rocks and bushes to larger ones like buildings and cars. From a
model sheet, a Modeler should have all of the information necessary to accurately model the asset so that it is consistent with the game’s visual style. Concept Design creates model sheets rapidly, producing between ten and twenty sheets per week depending on the team’s size.

Outside of model sheets, the Concept Design team also is responsible for the creation of user interface (UI) graphics and often collaborates with the Programming team on UI designs. User interface graphics include menus and sub-menus, in-game inventory systems, and head-up displays (HUDs). In addition, Concept Design provides the designs and artwork for any promotional materials released for the game, including websites, posters, advertisements, box art, and manuals.

Concept Design often assists the Level Design team in visualizing their environments by designing blueprint layouts of the levels based on specifications detailed in the game design document. In addition, Concept Design often works with Level Design in the creation of level paint-overs (see High Noon, Semester Two, Concept Design later in this chapter). A level paint-over is a collaboration between Level Design and Concept Design; a Level Designer takes a screenshot of a portion of their level that they feel needs a design overhaul and gives it to a Concept Designer. The Concept Designer then paints over the screenshot, adding environmental embellishments and structural changes to enhance the visual appeal of the level. These changes are then sent to the Level Design team, who implement the suggested changes.

If the game calls for cinematic quality images, it is the duty of the Concept Design team to provide professional, high resolution graphics that adhere to the game’s established style. High Noon, for example, featured an intro cinematic of stylized,
sequential images presented in a pan-and-scan format with accompanying custom music and narration. The sequence was storyboarded by the Director and passed to the Concept Design team, who in turn produced high-quality images that, when strung together, illustrated a complicated back story and provided the player with their first glimpse into the rich *High Noon* world.

**Modeling Team**

Formally, the Modeling team is housed under the Art Assets team, which encompasses both major art asset producers: the Modeling and Concept Design teams. The Modeling team is responsible for the creation of 3D art assets to be used in-game. On a given Chico State Game Studios project, the Modeling team will generate anywhere from one-hundred to three-hundred custom 3D models, depending on the size of the team and requirements of the project.

Modelers are tasked with creating a 3D model from model sheets produced by the Concept Design team. Specifications on model sheets include: physical description (presented in photographs and sketches by the Concept Designer), color palette, texture, scale, and context of the asset. From these model sheets, the Modeler creates a 3D model in his or her modeling package of choice within polygon requirements specified by the Modeling Technical Director. 3D assets for video games often come with strict polygon restrictions due to the number of assets on screen at one time in-game affecting the speed at which the game engine renders the environment. An excess of polygons on-screen at one time can bog down a game engine, making the game unplayable.

Once the model is created in a 3D modeling package such as *Lightwave* or *Maya*, the Modeler lays out a UV map in preparation for texturing. UV maps created for
video games must be efficient, wasting little space on the map area and overlapping UV’s when possible to ensure high resolution texture projections on models in-game. After the UV has been created, it is checked for distortion by using a checkerboard projection. Distortion is undesirable in UV-maps, as they cause textures to project incorrectly onto the models. Places where the checkerboard warps are indicative of UV distortion, and must be corrected before the model moves on to being textured.

Texturing begins with taking a snapshot of the geometry as it is laid out on the UV map and importing it into Adobe Photoshop. The Modeler then paints the texture onto the geometry screenshot, periodically saving their progress and projecting it onto the model to check that the flat, two-dimensional texture is lining up properly on the three-dimensional model. The textures for High Noon and D.A.V.I.S. are generally high quality, requiring the creation of detailed, imaginative, and often near-photographic quality images that take anywhere from one to twenty hours to complete. Additionally, some Modelers will create their own normal and specularity maps, alleviating some of the burden on the Technical Art team by providing pre-made maps for import.

After the model is textured, the Modeler then checks its scale in the Unreal engine by importing it using Unreal Editor. While the model is being checked for scale, the Modeler also verifies that the engine is lighting the mesh properly and that no modifications to the geometry need to be made before the model is sent off for approval.

Animation Team

As their name implies, the Animation team is responsible for all moving elements of a Chico State Game Studios production. During High Noon, the Animation team was tasked with creating looping, seamless animations for the weapons and enemies
featured in the game. The average *High Noon* weapon had seven animations attached: equip, unequip, fire, alternate fire, reload, idle, and long idle. Enemies for *High Noon* had varying numbers of animations, but most possessed a walk, run, attack, idle, long idle, and death animation. These animations had to be designed in such a way that they ran seamlessly together, so that when they were dynamically called in-game they looked natural and provided the player with appropriate visual feedback.

The Animation team is also responsible for animating any main CG characters, such as *High Noon*’s villain, Boz Novak. While Boz was for all intents and purposes an enemy, possessing all of the standard animations such as walk and die, he also required more sophisticated animations, including a monologue at the beginning of the endgame boss battle, taunting, and other visual cues to communicate to the player the current stage of the battle.

Weapon animations for *High Noon* and *D.A.V.I.S.* were created using *Autodesk Maya* and the *UDN ActorX Plugin* for digesting keyframe animations into a format recognizable by the *Unreal* engine. Once imported into the *Unreal* engine, animations were checked for proper playback and naming conventions, at which point they were passed to the Programming team to be assigned to the weapon via code in the custom weapons classes. Enemies were completed in a similar fashion.

The Animation team has one of the most crucial roles for a successful project: visually communicating to the player such important in-game elements as the presence of enemies and the rate at which weapons are firing. It is the duty of the Animation team to bring life to the characters and in-game environment, immersing the player and heightening the gameplay experience.
Outside of physically animating in-game meshes, the Animation team is also responsible for researching all moving elements to be implemented in-game. For example, during production on *D.A.V.I.S.*, the animation team did extensive research on *Unreal PhAT*, the visual physics modeling tool built into *Unreal Editor*. The Animation team also participates in the creation of in-game cinematics and scripted animated events, where applicable.

**Level Design Team**

Level Designers for Chico State Game Studios are tasked with creating the game world with which the player interacts. With the Concept Design team, Level Designers design the layout of the game world, blueprinting such things as landmarks, object placement, and scale. Level Designers work exclusively with *Unreal Editor*, building their worlds using the development tools provided by *Epic* and packaged with the *Unreal Tournament 3* game. The Level Design team is usually met with extremely steep learning curves as they familiarize themselves with the complicated and often temperamental game engine and development environment.

During *High Noon*, Level Designers were not required to learn many of the visual programming tools that come with the *Unreal 3* engine, which was largely an artifact of the pipelines used during the *Unreal 2004* implementation of the game. Rather, it was the duty of the Programming team to implement any custom behaviors, triggers, or scripts for the levels. Because of how burdensome this was for the Programming team and how much it deviated from practices used in industry, significant changes were made to the responsibilities of the Level Design team during the start of production on *D.A.V.I.S.*
By *D.A.V.I.S.*, Level Designers were not only responsible for the creation of the playable world in the game engine, but also for creating simple game scripts. Flickering lights, opening and closing doors, basic triggers, and simple cinematic events where camera control is removed from the player and focused elsewhere were among the items for which Level Design became responsible. In addition, Level Designers work with the built-in *Unreal* particle editor for creating fire, smoke, and other digital particulates for use in the game environment.

Level Designers commonly create paper versions of their digital levels before translating them to their electronic counterparts to check for playability, flow, scaling, and rudimentary aesthetics. These paper levels can be used in preliminary playtesting before the game reaches any sort of electronic prototype, ideally cutting down on game errors during production when major fixes are much more costly. From the paper levels, Level Designers create *BSP mesh* versions of their levels to further check scale, walking through the environment in-game to get a feel of how large the world will seem to the player. *BSP*, or *binary space partitioning*, allows the *Unreal* engine to identify negative and positive space as defined by primitive shapes like cubes. Using BSP geometry, Level Designers carve a rough representation of their level into the game world, allowing the team to see the draft of the level before it goes into full production. During this time, the Level Design team works closely with the Concept Design team to maintain visual consistency between the planned assets and the in-game world.

Of all of the Chico State Game Studios teams, the Level Design team is most dependent on the other sub-teams during the game creation process. Level Design is responsible for the placement of all in-game assets, including:
- Static meshes
- Animated (skeletal) meshes
- Enemies
- Weapon, ammunition and health pickups
- Terrain
- Lights
- Skyboxes/Skydomes
- Particles
- Music and sound effects

Because these assets are made by other sub-teams, Level Designers must wait for assets to be produced and cleared as production-quality before placing them in their levels. Therefore, the bulk of Level Design’s work is offset from the rest of the team, creating a potential problem for the production pipeline. While they are waiting for assets to be produced by the Concept Design and Modeling teams, Level Designers employ a method of design adopted by the Chico State Game Studios team affectionately referred to as “companion cubing” (see Chapter IV, New Chico State Game Studios Design Strategies, The “Companion Cube” Approach to Level Design). Using a generic cube with a checkerboard texture applied to it, Level Designers create stand-in static meshes for all of the assets that they project that they will need to complete their levels. By doing so, Level Designers are able to accurately pre-visualize their levels, ensuring that they are faithful renditions of the Director’s vision even before any art assets have been produced for the game (see Figure 4.2).
Figure 2. The “Companion Cube” reconstructed church and the final reconstructed church, assembled by level designer Nathan Smart.
As the Level Design team begins to integrate art assets into their levels, they perform preliminary playtests to verify that the level has no major flaws such as collision or lighting errors. They check the levels for appropriate scale and flow, ensuring that the player has enough visual information to progress through the level at a reasonable pace. In addition, the Level Design team works closely with the Director to schedule frequent and iterative playtests to incorporate player perspective in all design decisions made by the team. Playtesting starts as soon as the level is playable, preferably during the “companion cube” phase of building.

**Programming Team**

The Programming team is responsible for some of the most complicated and crucial tasks on the project: to implement all game functionality while ensuring an efficient and reliable product. Programmers are also responsible for implementing all functionality being used to make the game that is not native to the engine. In the case of *High Noon*, it was the Programmer’s duty to create custom classes for weapons, enemies, health pickups, ammo pickups, and weapon pickups per the specifications of the *High Noon* design document. To do this, the Programming team had to spend a significant period of time learning *UnrealScript*, the scripting language native to the *Unreal* engine. Additionally, they had to research the existing class hierarchies in the published *Unreal Tournament 3* game to develop strategies for implementing the custom *High Noon* game functionality.

All weapons in *High Noon* reloaded when their ammunition was depleted, unlike the weapons native to *Unreal Tournament 3*, which can’t be reloaded and instead simply do or do not fire depending on the amount of ammunition that the player
possesses. Because of this, reloading code had to be written by *High Noon* Programmers from scratch, calling custom animations created by the Animation team and keeping running tallies of the player’s in-clip ammunition versus their carried ammunition. In addition to reloading, all *High Noon* weapons had to interface with the other animations created by the Animation team, call the appropriate meshes and surface materials, and display at the proper scale and position on-screen. Each of *High Noon’s* nine custom weapons had to be meticulously implemented, ensuring that they functioned properly and interfaced with the many art asset variables needed to complete the weapon.

Much like the weapons, enemies in *High Noon* also required detailed custom functionality, but in addition had a significant added layer of complexity. Native enemies in *Unreal Tournament 3* are not driven by unassisted artificial intelligence, but rather use a technology called path nodes to detect where they should go in a given level. *High Noon* enemies were designed to detect the player’s position and draw paths to their location in order to attack. If the artificial intelligence was unable to detect the player, the enemies default to a patrolling state until the player is in close proximity. Each of *High Noon’s* seven enemy types had unique behaviors according to their physiology: Steam-Powered Grasshoppers hopped at an aggressive pace toward the player while shooting shoulder mounted rockets and had a generally long range of sight, whereas the Mechanical Spiders had significantly shorter range of sight and lunged at the player to attack.

*High Noon* also required custom user interfaces for not only in-game menus, but for special communication terminals that allowed the player to communicate with non-player characters in the game for delivery of quests and story. These user interfaces
were graphically designed by the Concept Design team, but their implementation was wholly the responsibility of the Programming team. Graphics provided by the Concept Design team had to be segmented and placed seamlessly in programmed in-game windows, buttons assigned and positioned, and dynamic content called and displayed appropriately according to in-game dialogue. Additionally, conversation trees had to be coded into the game, enabling the non-player characters to deliver appropriate sets of spoken lines in response to player interaction.

Many of *High Noon*’s features had to be created from the ground up simply because necessary functions did not exist in *Unreal Tournament 3*. For example, loading and saving games was a significant undertaking on the part of the Programming team, as *Unreal Tournament 3* lacked any sort of framework for saving games as *High Noon* needed them saved. The Programming team developed new standards for data storage and retrieval, allowing the player to stop and resume play at will.

For production on *D.A.V.I.S.*, the Programming team’s primary function was the implementation of in-game puzzles, which were essentially small mini-games with simple goals such as solving classic riddles or inserting a correct number of keys.

**Audio Team**

The Audio team was added to the Chico State Game Studios organizational structure during the third semester of *High Noon*’s production. In previous Chico State Game Studios productions, audio assets were largely created in an unstructured manner and treated widely as afterthoughts in the game’s creation. Audio is an increasingly important aspect of effective game development. Video games are by nature multimedia experiences, much like television and film. Without effective audio, most video game
projects feel like they are missing a layer of interactivity, resulting in a disconnect between the player and the game.

Audio is largely a post-production element, as it is difficult to create customized music and sound effects for assets and game worlds that do not exist. However, an Audio team can work during the preproduction semester and the first semester of production by using the time to experiment with different audio styles, eventually developing an audio template to base future work off of during the life of the project.

The Audio team often works independent of the majority of the Chico State Game Studios team because of the drastically different nature of their job on the project, supplementing the visual elements presented in the game. They are responsible for creating all music, sound effects, and post-processing standards for the game’s audio production, as well as converting audio into readable files for the game engine. In addition, the Audio team assists in the recording of any in-game dialogue or other special audio demands that the project may have.

Writing Team

Until *High Noon*’s third semester of production, writing was largely completed in-house by the Director. However, *High Noon* featured iconic characters, often requiring lines of dialogue for delivery of plot and active quests. Not only did writing become too cumbersome for the Director to complete alone, but the quality of writing suffered greatly simply because the Director lacked the necessary expertise.

Experts from the English department at California State University Chico were brought on to generate ideas, identify plot holes, develop characters, and write
dialogue for *High Noon*. Additionally, the *High Noon* writers revisited existing in-game cinematics, improving the existing writing and unifying the spoken and written tone with the visuals. They were so successful that writers were brought back on the production of *D.A.V.I.S.* to assist in writing in-game materials such as found articles and journal entries, inventory item flavor text, in-game cinematics, and dialogue.

Though a dedicated Writing team may not be necessary for all types of potential Chico State Game Studios projects, it is recommended to have writers involved as part-time staff for games with important character and/or plot elements. The success of *High Noon* and *D.A.V.I.S.* rests on the effective communication of their rich worlds and characters. Without on-staff writers to unify the characters, dialogue, and plot, both stories risked being lost in production or being treated as afterthoughts in favor of creating a technically functioning game.

**Technical Art Team**

The Technical Art team was introduced into the Chico State Game Studios organizational structure during *High Noon*’s third semester to account for the increase in complexity of surface materials in the *Unreal 3* engine. For Chico State Game Studios, Technical Art includes the creation of complex surface materials by linking nodal values into various surface channels in *Unreal*’s material editor. This complex process requires an understanding of various next-generation texturing techniques and of visual editors for sophisticated, multi-channel surfaces.

During the *Unreal 2004* implementation of *High Noon*, models were imported into the engine with their diffuse textures, which were assigned to one another in the *Unreal Editor*, and then immediately placed in the game. The implementation of 3D
assets was very simple, and did not require much more than a team member working part-time to ensure that asset packages were made and passed to the team. With the transition to Unreal 3 came a significant change in the manner in which 3D assets were implemented, largely because of the increase in complexity of next-generation surfacing techniques. Many 3D assets now require an assortment of specialized maps, including alpha, normal, and specularity, in addition to the previously required diffuse map. No longer a quick afterthought, asset importing and package building now takes hours. In addition to the 3D mesh, between one and five additional image files, or maps, must be imported. These maps are often high resolution, ranging from 512 x 512 pixels to 2048 x 2048 pixels, which have much larger corresponding file sizes, resulting in long import and load times. After the mesh and maps have been imported into the Unreal engine, materials must be set up through Unreal 3’s node-based material editor and applied to the in-game mesh. To account for these new technical art requirements, a team of specialists was established to study the material editor and learn best-practices for creating effective in-game materials.

A Technical Artist receives completed 3D models and textures from the Modeling team and imports them into the game using the Unreal Editor. From the supplied textures, the Technical Artist generates specialized maps to create channels for normals, specularity, and transparency when applicable. The Technical Artist is also responsible for determining if a model requires any sort of specialized properties such as emissive or animated material channels.

The preponderance of a Technical Artist’s time is spent using the Unreal Editor, and generally they do little physical art generation. Rather, they create formulas
using multiple surface channels and various input values to create convincing surfaces such as chrome, neon, rock, et cetera. The Technical Artist works closely with the Director, Assistant Director, and Art Director to ensure that their design decisions are consistent with the visual direction of the game.

When in-game materials are finalized, the Technical Artist creates game packages and sends them to the Level Design team for placement in their levels. A package is a collection of game models and their assigned materials, scaled appropriately and organized into groups that follow naming conventions established by the Director and Assistant Director. Packages are regularly updated throughout the life of the project, resulting in a massive asset library for use in the Level Design process.

Preproduction Team

Weak preproduction on previous Chico State Game Studios projects prompted the introduction of the Preproduction team into the Chico State Game Studios organizational structure. The Preproduction team is responsible for designing the game in its entirety, including authoring the game design document, establishing the game world and gameplay, and developing any characters, puzzles, or other critical game elements.

The Preproduction team is the only sub-team to work on the game before production formally starts. Largely external, the Preproduction team is small, usually consisting of no more than six senior-level students with significant experience in game design and production. It is recommended that the members of the creative committee have a solid working knowledge of the major teams on a Chico State Game Studios production: Modeling, Animation, Programming, Level Design, and Concept Art to ensure that design decisions are within the scope of each team’s abilities.
Preproduction work takes place in parallel with the final semester of production on the current game project, preparing the new game for full-fledged production. The Preproduction team will continue to meet throughout the production life cycle to ensure that any necessary changes to the game’s plot, characters, or quests remain consistent with the overall vision for the project. In short, the Preproduction team not only authors the formal and dramatic elements of the game, but also provides vital quality assurance throughout the projects duration.

The Production of High Noon: A Historical Recount

Chico State Game Studios’ current production pipelines were developed over the course of three semesters and two projects: two semesters of *High Noon* using the *Unreal 2004* engine, one semester of *High Noon* using the *Unreal 3* engine, and one concurrent preproduction semester of *D.A.V.I.S.*, the successor to *High Noon*. At the end of each semester, production pipelines and team organizations were evaluated for effectiveness and modified as necessary. Generally, new experimental pipeline ideas were introduced into the organization each semester, and then evaluated between semesters for possible use in subsequent semesters. Designing what would become the working production standards for Chico State Game Studios in this iterative fashion proved to be very successful, allowing for a level of complexity and accuracy in the production pipelines previously unattained by the organization. The pipeline created from research conducted during *High Noon* and *D.A.V.I.S.* is designed to serve as a template for future Chico State Game Studios productions, integrating industry-standard processes with processes tailored for Chico State Game Studios as a unique student organization.
High Noon, Semester One

The idea that inspired the game *High Noon* was selected in the spring of 2007 from a pool of six game ideas presented by students to the Chico State Game Studios Executive Committee. Shortly after the game was selected, the Director was appointed and announced. According to precedent, all preproduction was to be finished by the Director prior to the beginning of the semester following the game’s announcement as the next Chico State Game Studios project. Preproduction at this time included:

- Designing the rules and gameplay mechanisms used by the game.
- Establishing the game world: story, characters, and back story.
- Authoring the initial design document.
- Creating a cumulative production pipeline.
- Establishing team organization.
- Recruiting talent as necessary for game production demands.

Like *Bronze Age* and *Viathan* before it, preproduction on *High Noon* began the summer before the game was scheduled to go into full production. *High Noon* was based loosely on a treatment describing a multiplayer action “death match” game in specialized, Western-themed maps. Ultimately, *High Noon* became a single-player action-adventure game, featuring non-player characters and linear, quest-driven gameplay. While many changes were made from the original game idea, *High Noon* remained in the Western genre. Changes to the game’s structure introduced Steampunk elements to ensure a unique visual style and whimsical nature to the plot and characters. After a game type and genre were formally selected by the Director, the first draft of the game design document was authored, detailing core game rules, major game levels, back story,
characters, desired visual direction, weapons, and preliminary enemy ideas. Many aspects of the game were not explicitly designed at this time to allow for creative input on the direction of the game from the team (see Chapter IV, *New Chico State Game Studios Design Strategies, Emphasis on Story*). This early version of the design document offered guidance to the team, but made it clear that elements of the game would change as designs were created and new ideas generated. Even characters that became cornerstones for the identity of *High Noon* were only outlined in the preliminary design document; for example, the description for Boz Novak, *High Noon*’s iconic villain, was purposefully vague, designed to encourage development of his character and motivation:

Boz Novak is an opportunistic land baron who is intent on adding Sutter’s Mill to his growing empire. A model villain, Novak is dark-hearted, cut-throat, and only interested in one thing: money.

*High Noon* is designed to not only play with, but embrace the cliché. Feel free to do this with Novak. He’s eeeevil. Yes, even with all the extra “e’s”. Does he need to be a deep character necessarily? Absolutely not. He provides a threat to Sutter’s Mill and to Jekyll, he’s the primary reason for the utilization of the Hyde parasite (see the Story Section), and he’s (quite naturally) the final boss. Boz Novak is designed to provide conflict, and anything else he manages to do along the way is gravy.

Have fun with Boz. Give him the evil goatee and the menacing top hat. The design of Boz is fairly open as long as it communicates one primary idea: that he is, without a doubt, the antagonist. (Thayer, 2008, p. 28)

With the work of the Concept Designers on *High Noon*, Boz Novak eventually developed a personality, detailed back story, motivation, logo, and small cult following at Chico State Game Studios. The working details of *High Noon* were designed mostly in this fashion, which generally served as a very successful and creative game development method for Chico State Game Studios. However, using such a free-form method of design was not without its hazards, which brought additional changes to the production pipeline in later semesters.
The first semester of *High Noon* marked a number of significant changes to the production pipeline used by Viathan, the Chico State Game Studios project preceding *High Noon*. Fundamentally, *High Noon* started with a significant structural change for Chico State Game Studios; in previous semesters, Chico State Game Studios functioned as a credit/no credit special topics course. During *High Noon*, Chico State Game Studios became APCG 495: *Advanced Production*, a 3.0 unit graded course with formalized meeting times and a scheduled lecture period. This change occurred on the recommendation of previous Directors, who stressed that the loose structure of a special topics course was not enough to reinforce the professional environment needed to generate the large-scale productions Chico State Game Studios sought to create.

The legitimacy that came with an official course number and formally published meeting times allowed Chico State Game Studios to introduce new levels of structure to the organization at the start of *High Noon*. The first night of the class, the Director distributed a syllabus (see Appendix D), detailing important information regarding course expectations and the requirements of participating in a large-scale video game project. Grades were clearly defined as a combination of Mid-Term and Final Self-Evaluations (see Appendix I), Director evaluation, and attendance. Attendance became a requirement of the Chico State Game Studios course, which explicitly stressed in-class team communication and collaboration over working independently out-of-class. Team members were required to utilize various communication methods outside of class; including telephone, e-mail, and instant messenger technology, as well as provide their personal contact information to a team contact information document which was housed
on the class website. This document helped teams stay in close contact with one another over the course of the semester.

Team structure during the first semester of *High Noon* was modeled after the format used by Chico State Game Studios for the *Bronze Age* and *Viathan* projects, and then modified to suit the needs of *High Noon*’s production. The Chico State Game Studios team was composed of one Director, one Assistant Director, and five sub-teams: Modeling, Concept Design, Animation, Level Design, and Programming. Modeling and Concept Design teams were experimentally merged into one Art Assets team, but were frequently managed as sub-teams below the Art Assets team designation. Team responsibilities were defined as follows:

- **Executive Producer**: Faculty member responsible for the class, assigns letter grades, serves as final say in all inner-team conflicts, maintains weekly communication with the Director to ensure the project is running smoothly.

- **Director**: Creates the original game idea, structures the classroom environment and teams, manages all aspects of production (preproduction, production, and postproduction), recruits new talent, manages day-to-day classroom operations, sets milestones and arranges schedules, manages sub-teams, mitigates team conflicts, authors game documentation, manages assets, creates the game production pipeline, establishes art and production standards, serves as a surrogate team member for any sub-team on which worker shortages or technical difficulties arise.

- **Assistant Director**: Responsible for researching game technologies necessary for implementing the specifications of the game project.
- **Modeling (Art Assets):** Responsible for the modeling, UV mapping, and texturing of 3D game assets including enemies, characters, weapons, buildings, and props.

- **Concept Design (Art Assets):** Responsible for the creation of model sheets for 3D asset production, character designs, environment designs, back story illustration, character and story development, weapon designs, and enemy designs.

- **Animation:** Rigging and animating custom in-game characters, enemies, and weapons; ensuring animations import properly into the *Unreal 2004* engine.

- **Level Design:** Designing each of the four game levels, including quests, weapon pickups, enemy spawn points; layout the game world in the *Unreal 2004* engine, providing asset demands to the Modeling team, placement of lighting, particle effects, and blocking volumes.

- **Programming:** Implementation of all game functionality, including user interfaces, in-game HUD, custom artificial intelligence, weapon and enemy implementation, game balance, core game functionality, and quest implementation.

Once the sub-teams were established, I selected a Technical Director for each sub-team. At this point in the Chico State Game Studios organization, a Technical Director was considered to be the main point of contact on a given sub-team for any issues during production, including quality control, technical difficulties, et cetera. Beyond the responsibilities to their team, a Technical Director was required to meet with the Director and Assistant Director once a week to discuss pertinent team issues and ensure that the project was progressing satisfactorily. Outside of administrative duties, the Technical Director also served as a content producer like the rest of his or her team.
Technical Directors during the first semester of *High Noon* were not tasked with any sort of quality control or asset management, with that responsibility falling exclusively to me as the Director, which was re-evaluated in later semesters.

In addition to the five major teams, I introduced to *High Noon* a Cinematics Team to account for planned cinematic movies that both introduced and concluded the game. Composed of three artists and managed as an independent team, the Cinematics Team produced high-quality, high-resolution images to be used in a simple pan-and-scan movie, set to custom music and narration that delivered important aspects of the *High Noon* back story to the player. The Cinematics Team was not managed by a Technical Director, instead reporting to me for input, assignments, and creative direction.

Chico State Game Studios policies during *High Noon* centered largely around maintaining communication and stressed heavily the value of both a creative team environment and its impact on the creation of a good game. *High Noon* was managed largely using a free web application for group management produced by Google called a *Google Group*, which allows members to easily e-mail one another, post valuable files, and maintain group discussions on important aspects any project. The *High Noon Google Group* became a repository of vital information about the project and provided a central means of communicating with the *High Noon* team.

The structure of the Chico State Game Studios course during *High Noon* was designed to emphasize inter-team communications on both global and local levels. Activity times, held twice weekly, were focused primarily on sub-teams working closely together to accomplish specialized, designated tasks, and served as a venue for local communications between group members. Lecture periods, held once weekly, were
reserved for “weeklies,” or a weekly wrap-up presentation of the accomplishments of each team for the week. The use of regular weeklies reinforced global team communication, ensuring that all sub-teams were aware of the current tasks and accomplishments of the other sub-teams. Furthermore, weeklies allowed team members to watch the game as it progressed into final product.

I also introduced the concept Mid-Term and Final Self Evaluations (see Appendix I) during the High Noon project. As the name implies, these evaluations were administered at the mid-term and end of the semester to evaluate student performance, but most importantly to collect data on how to improve the High Noon project and Chico State Game Studios as an organization. The Self-Evaluation proved to be a valuable means of communication between the Director and the teams, creating a venue in which to express concerns, suggest new ideas, and even reaffirm that the structure of the organization was satisfactory.

The High Noon Semester One Production Pipeline

The production pipeline used largely for the first semester of High Noon was experimental at best, based off of a limited knowledge of the Unreal 2004 engine and existing production pipelines. Due to a lack of standardized documentation from previous projects, Chico State Game Studios had few production standards to translate into the new High Noon project. As such, many of the production standards and practices used on the first semester of High Noon were flawed, often resulting in bottlenecks and stalling production.
The initial pipeline was organized by teams, as illustrated in Figure 3. Expected pipeline dependencies, team allocations, and major tasks were explored. Assignments and milestones were created based off of this chart, directly corresponding to establishing due dates and expected completion for the project.

To complement the new production pipeline, I established the use of the Assessment Exercise (see Appendix E) as the first assignment of a semester production cycle. Created in response to the disparate levels of experience of team members on a given Chico State Game Studios project, the Assessment Exercise provides the Director with a standard method of quickly assessing the skills of his or her team. This allows for a more accurate projection of asset flow and ensures that team members are assigned work within their existing skill set. In addition, assessment effectively allows new Chico State Game Studios students to gauge the difficulty level of the class and determine if their schedule will allow for the intense time commitment.

**Concept Design**

As previously stated, when *High Noon* went into production in the fall of 2007, very little of the game had been established, either stylistically or functionally; the design document instead contained suggestions to the team on how to design various aspects of the game. Therefore, a large responsibility rested on the heads of the Concept Design team to establish the look and feel of the weapons, enemies, characters, and game world quickly so that the rest of the team could begin implementing the game. Generally in industry, the tasks for the Concept Designer are encapsulated in preproduction, with relevant images and designs generated before the game goes into full production. These images are delivered to the team as a point of reference for the project,
Figure 3. The proposed *High Noon* pipeline for semester one.
with all sub-teams using the designs and illustrations as inspiration for their duties. At Chico State Game Studios, with preproduction occurring between academic terms, it was generally very difficult to contract students to work on the current game project with any sort of speed and/or reliability, and so it was rare that finalized art direction or asset designs were completed prior to the start of production.

Not only that, but the Concept Design team structure and practices are an area where Chico State Game Studios greatly deviates from game companies in industry. Chico State Game Studios is operated in a classroom environment and therefore does not select the team working on a given project. Rather, students elect to sign up to work for Chico State Game Studios each semester as their schedules allow, and generally a significant percentage of them are artists seeking Concept Design experience. As a result, Chico State Game Studios typically employs a full Concept Design team through an entire production cycle, and commonly has little control over exactly how many Concept Designers there are in a given semester. For this reason, the Chico State Game Studios Concept Design team operates year-round, producing not only concepts for the game, but promotional materials, box and manual art, back story illustrations, cinematic designs, and on-demand concept revisions throughout a production cycle.

Like the rest of the High Noon sub-teams, the Concept Designers’ semester began with an Assessment Assignment (see Appendix E), tailored to their team to assess their comfort with the technology and ability to deliver production-quality content on short deadlines. Concept Designers were allowed the choice of focusing on environmental or character concept art. Concept Designers choosing to focus on environmental concept art were tasked with developing concepts for one of two High
*Noon* levels: Sutter’s Mill or the Reconstructed Church. Character Concept Designers were given the choice of providing designs for either Boz Novak or one of his robots, the Mechanical Spider. Like with the rest of the sub-teams, it was understood on the Concept Design team that successful concepts would be used in the final implementation of the game. In fact, it was from the assessment exercise that concepts used in-game for Boz Novak and his Mechanical Spider originated.

Concept Design was also responsible for the development of level concepts, characters, and environmental (Second Tier) assets, as well as game-specific items such as health and ammo. Concepts generated included: model sheets, designed to send directly to a Modeler for use in making 3D assets (see Appendix G); level lighting and color treatments; character sheets; and back story cinematic illustrations. Concepts created by the Concept Design team were then passed to the Modeling or Level Design teams to assist in asset and game implementation. Generally, Concept Design marks the beginning of an asset’s journey through the production pipeline, so rarely did the Concept Design team directly feel the effects of any pipeline bottlenecks.

**Modeling**

During the first semester of *High Noon*, the Modeling team consisted of five team members: the Technical Director and four junior Modelers. Because of the exceptionally small team size, combined with the fact that no real precedent existed for modeling production, the pipelines used for Modeling changed frequently as the team experimented with different production standards.

The duties of the Modeling team were largely dependent on the tasks completed by the Concept Design team. While Modelers did at times work without
existing concepts in the creation of 3D assets for *High Noon*, most modeling tasks were first envisioned by the Concept Art team and then passed to the Modeling team for implementation. At first, this pipeline was extremely difficult to employ because of the fact that preproduction was practically non-existent on *High Noon* when it went into production. Because no concepts existed at the beginning of the first semester, the production pipelines used by the concept-dependent teams (Modeling, Animation, and Level Design) had to be offset from Concept Design’s pipeline by about three weeks to allow for the rapid generation of asset concept pieces.

Modeling used the three week waiting period as a time for training and research. Assessments were assigned to gauge each team member’s skill level and comfort with the technology used in the class. Assessments (see Appendix E) for Modelers included the design and construction of a Steampunk-inspired Western building or structure, selected from a list provided on the assignment. If the model generated for the Assessment was considered to be of production quality, it was understood that it would be used in the game.

After the completion of the Assessment Assignments, Modelers were given individualized assignments by the Director ranging in difficulty from lower-priority scenery pieces to higher priority weapons and enemies. Deadlines during the first semester of *High Noon* were very flexible, as it was unclear how long it would take for Modelers to design, model, UV map and texture any of the art assets. Modelers were asked to complete assignments in a “reasonable” amount of time, which usually meant one to two weeks depending on the complexity of the model. Because of this, the flow of assets was generally unpredictable in *High Noon*’s first semester.
Models created during *High Noon*’s first semester of production were not generally closely checked for quality and engine-appropriateness largely because there were no preexisting quality standards for Chico State Game Studios projects. All parties, including the Director, were experimenting with *Unreal* technology for the first time, resulting in a large number of mistakes, technological misunderstandings, and delays in production. Because there were no set asset guidelines, many resulting models exceeded reasonable polygon limits, contained intersecting geometry, or had other flaws that prevented the game engine from processing the model properly. Though standards were eventually developed and enforced during the first semester, a large number of models created in the beginning of the semester had to be revisited due to the fact that they were unusable in the *Unreal 2004* engine.

As the semester went on, we were able to better predict reasonable polygon restrictions on the assets generated for the game. Modeling assignments were delivered with strict requirements in terms of number of triangles used, distance between vertices (long distances between vertices risk being lit improperly in-game because *Unreal* uses vertices to calculate lighting), and rules regarding intersecting polygons, which caused the engine to light meshes inconsistently.

Because there were no documented texturing standards for Chico State Game Studios games using *Unreal 2004*, I spent a great deal of time painting high-priority assets to ensure they were of appropriate production quality until standards could be established. During this time, structured painting techniques were more or less accidently created and *High Noon* developed a defined, reproducible art style. This art style was
taught to the Modeling team over time, becoming more and more specific as various art techniques were explored for the game’s visual identity.

High Noon’s 3D assets were categorized as “First Tier” and “Second Tier” for the purposes of prioritization and categorization of asset difficulty. First Tier assets were generally higher priority assets, requiring higher levels of detail, in particular polygon count and texture resolution, because of their proximity to the player or importance to the game. First Tier assets were usually reserved for senior modelers because of their high level of complexity and importance to the game. Second Tier assets were considered to be primarily environmental, ranging in size and importance from small, simple objects like rocks and crates to large, more complex objects like buildings and environmental landmarks. Most 3D assets produced for High Noon were considered Second Tier assets, with around three-hundred unique objects created by the end of the project.

Modelers in High Noon generally used the modeling package they were most familiar with to create their 3D assets. Software used included Maya, Silo, Lightwave, and 3D Studio Max. Most completed models were exported from their native 3D modeling package and imported into Maya, which had the most reliable UDN ActorX plug-in, and exported into Unreal Editor’s preferred format, ASE, or ASCII Scene Exporter format.

Texturing was achieved by taken UV snapshots from 3D models and painting over them in Adobe Photoshop. Most textures for the Unreal 2004 implementation of High Noon were painted using canvas sizes of 512 x 512 pixels, at 72 dpi. First Tier assets were larger, usually painted at 1024 x 1024 pixels and 72 dpi. Textures rarely exceeded 1024 x 1024 pixels during the Unreal 2004 implementation of High Noon. To
produce textures, Modelers mixed photographic references with hand-painted elements and arranged them according to the geometry projected on the two-dimensional UV map. Periodically previewing the texture in *Maya, Silo, or Lightwave*, the Modeler would verify that all painted elements were projecting correctly.

When models were completed, they were sent to the Director for integration in to 3D asset packages used by the Level Design team to populate their levels. Asset importation was relatively simple; meshes were imported into the game and checked for scale, and then their corresponding textures were imported and assigned. Generally, surfaces in the *Unreal 2004* implementation of *High Noon* did not possess any special properties such as specularity or reflection. Occasionally, alpha maps were implemented to simulate complex geometry with relatively inexpensive rendering requirements.

**Animation**

The Animation team was the smallest sub-team during the first semester of production on *High Noon*, consisting of one Technical Director and two dedicated team members. Animating elements for *High Noon* was frequently an iterative process, as none of the members of the Animation team had any experience animating for video game production, and documentation on proper techniques was largely non-existent. Most of the animations produced for *High Noon* were done so using trial and error, developing standards to use in the future as the project matured.

Like the other *High Noon* sub-teams the Animation team completed an Assessment Assignment wherein they animated a prototype for the Mechanical Spider. The Animation team’s assessment generally lasted longer than most of the other sub-team’s assessments because of the complex process involved in creating animations for
Unreal 2004. Though the animations themselves were simple, the act of ensuring that they could be strung together seamlessly while being dynamically called in-game was not. The Animation team spent a significant period of time establishing best practices for creating animations that imported and played in Unreal 2004 correctly.

Animations were produced at a slow pace during the first semester of High Noon because of the small size of the Animation team and the fact that all animated assets had to first be modeled and UV mapped before they could be sent to be animated. By the end of the first semester of High Noon, the majority of the in-game weapons had all of their basic firing animations created and working in-game. Reloading animations were not pursued during this semester, as custom code had to first be written to handle the non-native reloading functionality. Enemies, which were started after the completion of all of the weapons, were not sent to the Animation team in time to be animated by the end of the first semester.

Level Design

The Level Design team was responsible for creating the game world in which High Noon takes place. Each Level Designer during the first semester of High Noon was assigned to one of the four projected levels: Sutter’s Mill, The Reconstructed Church, The Pre-Collapse Graveyard, and the Train Depot. Before Level Designers began their electronic implementations, they were tasked with designing their levels using grid paper and narrative treatments for their Assessment Assignments, outlining all major level features such as player progression, weapon and ammo drop points, enemy spawn points, and sub-quests. The Level Design team was also asked to provide aesthetic treatments of their levels, including color palette, architectural themes, music, and sound effects.
Once their paper prototypes were approved, Level Designers began to experiment with *Unreal Editor* to familiarize themselves with the tools that they would use to build their levels. Once they reached a sufficient level of comfort with the basic workings of the software, they blocked out their levels using *binary space partitioning* (BSP) geometry. This allowed the team to visualize a primitive representation of each of the major *High Noon* levels, blocked out roughly for scaling purposes and defining level flow. Blocking out levels using BSP geometry was a relatively simple process, however, and Level Designers quickly found themselves waiting on the other *High Noon* teams before they could proceed with their own work.

**Programming**

Programmers on *High Noon* were tasked with implementing all of the features that *High Noon* would need to function as a game that were non-native to *Unreal Tournament 2004*. This included the custom head-up display (HUD), navigation and menus, weapons, enemies, cinematics, artificial intelligence, terminals for non-player character interaction, scripting and triggers, level transitions, and end-game conditions.

The duties of the Programmers were monumental compared to those of the rest of the sub-teams. Not only did they have to quickly familiarize themselves with all of the subtleties of making a game modification project work using *Unreal 2004*, but they had to implement on the fly, making sure that the other sub-teams had access to crucial code assets throughout production. To assist with this steep learning curve, Programmers were tasked with switching the ammunition types on two existing *Unreal 2004* weapons for their Assessment Assignments. This ensured that all Programmers knew how to navigate the existing code, compile new code, and scrub the *UnrealScript* API for class
definitions and information on hierarchies. Programmers were responsible for the design and implementation of all of the *High Noon* weapons, creating custom classes for each weapon that extended from existing *Unreal 2004* classes. Not only that, but Programmers were responsible for defining new, non-native weapon behaviors, in particular for complex weapons such as the Schmittou Smasher Pipe Wrench. Unlike *High Noon*, *Unreal 2004* does not have functionality for melee weapons, and all functionality for the *High Noon* wrench had to be designed and written from scratch.

Enemies were the Programming team’s most significant task during the production of *High Noon*, as *High Noon* enemies were greatly different from the enemy types native to *Unreal Tournament 2004*. In *Unreal Tournament 2004*, enemies are coded to search for elements placed by the Level Designer called *path nodes*, which tell the artificial intelligence (AI) how to navigate a level, where to find ammunition, and how to locate the player. Enemies in *High Noon* did not use path nodes, but rather were state-driven, idling or patrolling when the player was not in sight, and then attacking when the player came into view. Many sophisticated functions had to be written by the Programming team to ensure that the *High Noon* enemies functioned properly.

The Programming team was also responsible for the implementation of the game’s custom head-up display (HUD) and custom user interface (UI) elements such as the main game menu and corresponding sub-menus. Though the visual elements were designed by the Concept Design team, the Programmers had to ensure that the images imported properly into the game engine and dynamically updated when appropriate. One of the most significant UI elements that had to be designed were the Communication Terminals, which were accessed by the player to obtain updated information on current
quests, story, and non-player characters. These terminals traversed dialogue trees based on the player’s current quest and displayed important information through character dialogue and portraits on how to progress through the game.

As Unreal 2004 was not built for linear, single player games, all triggers and quest functionality had to be custom designed to progress the delivery of game story. Due to a significant error in the Level Design pipeline, the Programming Team was not able to focus on the implementation of custom triggers or level functionality until the end of the first semester. This allowed them little time to complete this extremely complicated task, and resulted in largely non-functioning level quest progressions by the end of the first semester.

External Teams

The first semester of High Noon included experimentation with the addition of external teams to Chico State Game Studios. Specialized Concept Designers formed an external Cinematics team to handle the requirement for in-game cinematic events that narrated complicated back story necessary to understanding the setting in which High Noon takes place. Additionally, a musician was brought on to experiment with music for game levels and the intro and endgame cinematics.

The Cinematics team worked exclusively with me to produce high-resolution, professional cinematic images that illustrated the major events leading up to the High Noon game. Over the course of the semester, the team of three produced fifteen images for the intro cinematic during the first semester, which were strung together and put to narration and music and used as the well-received opening for High Noon. The positive
results yielded by the experimental Cinematics team lead to other instances of external sub-contracting in later semesters of the *High Noon* project.

Though audio production was experimented with during *High Noon*’s first semester, it was widely unsuccessful due to the unstructured nature of its production. All audio produced during the first semester of *High Noon*’s production was largely unusable and had to be revisited in later semesters, and audio was not actively pursued again until the *Unreal 3* conversion.

**High Noon Semester One Production**

**Pipeline Observations**

**Concept Design, Modeling, and the Creation of Art Assets**

At the beginning of the *High Noon* production cycle, Concept Designers were tasked with quickly developing model sheets for the eleven original weapons and nine original enemy types associated with the game so that they could be sent through the modeling pipeline. Erroneously, concepts for weapons were created first, and were therefore sent first to the modeling team for modeling and texturing. While certainly difficult to implement and possessing a significant number of inter-team dependencies, weapons were in retrospect not the most complex and time-consuming element of *High Noon*’s design. Moreover, the fact that there existed eleven weapon types to be concepted and modeled over five Concept Designers and five Modelers ensured that the production pipelines for both teams would be tied up for many weeks, with each weapon taking about a week to concept and a minimum of two weeks to model and texture. This resulted in weapons taking at minimum three weeks for the Concept Design team to be free to
take on a new task, and an offset four-to-five weeks to complete the 3D assets before the modeling team would be available to create other game-critical assets. This is particularly a concern when a production semester consists of fifteen weeks, meaning that roughly 50% of the first semester of *High Noon*’s production was dedicated solely to the creation of weapons (see Figure 4).

**Figure 4.** Concept design and modeling dependencies.

Enemies, in fact, would have been a more appropriate first task for the Concept Design team. Had the pipeline been more carefully considered for the *Unreal 2004* engine, it would have been identified that the game on which *High Noon* is built, *Unreal Tournament 2004*, contains functioning weapons that are similar to the *High Noon* weapons that would have made excellent temporary stand-ins for the custom *High Noon* weapons. Enemies, however, are significantly more complex game elements, requiring sophisticated technical development in animation, physics, artificial intelligence, and collision, and do not have close counterparts built into the existing *Unreal Tournament 2004* game. By creating the enemies first, those many complicated dependent tasks could have started significantly earlier in the project’s development,
leading to more opportunities for playtesting and refinement. The *High Noon* weapons are so similar to the existing *Unreal Tournament 2004* offerings, in fact, that gameplay would have been near-identical had the original *Unreal Tournament 2004* weapons been used as placeholders during *High Noon*’s early development.

In the middle of the first semester of production, after the completion of First Tier asset concepts such as weapons and enemies, Concept Designers and Modelers were paired in an attempt to streamline the asset creation process. By dedicating a Concept Designer to a Modeler, it was hoped that the Modeler and Concept Designer would be able to create a more unified vision of art assets for the game. While the concepts and models created during this time were more accurate to the overarching vision of *High Noon*, this process did not increase the quality of models significantly enough to warrant using it in subsequent semesters, largely due to the fact that quality control of designs and 3D assets had not yet been formally introduced into the pipeline. Many concepts and models still contained inaccuracies and technical flaws by the time they reached the end of the pipeline, requiring revisions or complete redesigns and ultimately costing more time than was saved. The act of pairing Concept Designers and Modelers came close to solving the problem of quality control on assets created for the game, but ultimately it was the addition of the Art Director role to the Chico State Game Studios organization at the start of *D.A.V.I.S.* that provided a solution. The Art Director brought unification to the concepts being generated by the Concept Design team and their interpretation by the Modeling team.

In addition to generating concepts for production of in-game assets, the Concept Design team was used as a reserve unit for 3D asset texturing during the end of
the first semester of production. Because semester one of *High Noon* marked the first time Modelers were required to texture their own assets, concerns existed for the speed at which the Modeling team could generate the completed 3D assets needed for the game to function properly. Because of the strong artistic background that Concept Design provided, it seemed a logical choice to include them in the texturing process when the workload became too much for the Modeling team to handle alone. Moving Concept Designers to the Modeling team for texturing purposes also served as solution to a long running Chico State Game Studios problem: finding work for Concept Designers once all of the primary assets had been conceptualized.

Moving Concept Designers temporarily to the Modeling team in actuality caused a massive bottleneck in production that resulted in many assets that had to be touched up or completely repainted in the second semester of production. While Concept Designers were certainly adept at digital painting in terms of generating concepts, the skills and knowledge necessary for creating convincing digital textures were significantly different than those used to create convincing concept art. Because Concept Designers were introduced to the texturing pipeline so late in production, the learning curve proved to be too steep to create production-quality texture assets in the allotted amount of time. It became apparent that teams dropped too suddenly into an established pipeline could easily flounder, as most lacked the knowledge and skill sets to accomplish their tasks quickly and effectively.

**Level Design Meets Massive Bottlenecking**

The Level Design team faced the largest number of bottlenecks in the production pipeline in the first semester of *High Noon* of any sub-team. Though they
spent the first part of the semester working primarily on paper, they still were greatly
hindered by the slow flow of assets through the production pipeline. Most Level
Designers found themselves in production lulls while they waited for the other teams to
produce the assets they needed to complete their tasks.

Because it was believed at the time that level functionality such as quest
implementation and scripted events could not be achieved in levels that were not
aesthetically complete, the Level Design team did not even begin to think about quests
and how they affected the design of their levels until late in the semester. This was easily
the biggest flaw in the High Noon pipeline. When Level Designers should have been
finding ways to test their quests and level flow, they were instead waiting for assets to
trickle through the pipeline, placing them one at a time. Level Designers felt generally
that their hands were tied, and I was unable to come up with any suggestions on work
they could be doing besides modifying their lights, placing particles, and ensuring that
the levels were of appropriate scale.

Because of this, the implementation of quests was largely an afterthought
while we were distracted by asset production. By the end of the first semester, while High
Noon was certainly visually pleasing, it was largely unplayable outside of simply
exploring the levels. This had further repercussions through the project, as it kept it from
entering any sort of playtesting phase until the middle of the second semester of
production (see Chapter IV, New Requirements of Chico State Game Studios Projects,
Playtesting).

In High Noon, the defined duties of the Level Designer deviated most from
their industry counterparts. "The Level Designer . . . is responsible for all of the technical
details of the play space: layout, AI pathing, object placement, spawning position, encounter triggers, cameras, collision, cut scenes, and scripting” (D. Cross, Nihilistic Studios, personal communication, February 18, 2009). Semester one High Noon Level Designers, however, were responsible only for creating the visual world in which the game takes place, designing quests, designating enemy spawn points, placing weapon and ammo pickups, and environmental elements such as lighting and particle effects. Generally, Level Designers at Chico State Game Studios lack experience with the programming and animation skills needed to fulfill the full industry-standard role of Level Designer, and were therefore treated primarily as designers rather than implementers. Because of these reasons, scripted events, triggers, cameras, collision, and cut scenes were the responsibility of the Programming Team.

High Noon, Semester Two

Semester two of High Noon involved a significant number of adjustments to the previous semester’s pipelines based on their successes and failures. The game design document was revisited with a critical eye, and modifications were made based on the established designs from the previous semester and new knowledge of the game design process. Overall, the second semester of High Noon’s production was focused on completing the tasks that were not finished during the first semester, and getting the game as close to an Alpha release as possible.

All teams at Chico State Game Studios experienced changes in the pipelines used to produce assets due to an expanded knowledge of the Unreal 2004 engine and the fact that the team size had increased from eighteen to twenty-nine.
Concept Design

Many of the responsibilities that the Concept Design team had during the first semester of *High Noon* continued during the second semester, but new tasks arose as the team better understood the demands of the project. In addition, many of the major assets had already been concepted, meaning that the demand for model sheets and specific asset concepts declined significantly in the second semester. To keep the Concept Design team producing, new options had to be explored for the team.

During the first part of the second semester, the Concept Design team provided heavy assistance to the Level Design team by completing *level paintovers* (see Figure 5) to help Level Designers visualize their levels with more sophisticated placement of assets, relevant decoration, and increased ambiance. Level Designers provided Concept Artists with screenshots of portions of their levels that they felt were barren or needed renovations. Concept Designers painted over these screenshots in *Adobe Photoshop* or *Bauhaus Mirage* to suggest subtle visual elements or complete architectural overhauls. The completion and approval of level paintovers resulted in the creation of new assets and further population of the existing levels, giving them more character and visual appeal. These images proved to be exceptionally helpful to the Level Design team in alleviating design difficulties and greatly improved the aesthetic qualities of the *High Noon* levels.

Many Concept Design team members began exploring the *High Noon* mythology using visual images. Characters histories were explored, with Concept Designers creating images such as childhood photographs, newspaper clippings, and other back story related images. This material was distributed through the team,
Figure 5. An example of a level paintover, painted by concept designer Jade Reed.
increasing the understanding of the story, characters, and game world. These high-quality images took between two and four weeks to complete, depending on level of complexity, and were assembled into an art book for distribution through the team.

Concept Designers also contributed to the creation of decals, which are transparent textures that overlay other in-game textures to add personalization and randomness to game environments. Many members of the Concept Design team spent their second semester creating poster designs, which were applied to in-game buildings by the Level Design team using decals. These posters, designed to communicate backstory and game world mythology to the player through the environment, featured advertisements from fictional weapon manufacturers, ironic Steampunk references, and even messages from the game’s villain, Boz Novak.

Modeling

The Modeling team’s tasks for the second semester of High Noon’s production remained consistent with those of the previous semester. By the end of the first semester, enough 3D assets had been created to produce skeleton layouts of the four major levels, but it was clear that none of the levels were near populated enough to qualify as final levels. In addition, the majority of assets made for the first semester of High Noon were unfinished, and most possessed poor UV-maps and no textures.

One of the largest issues with production during the first semester of High Noon was the slow rate at which textures were being generated for models. In fact, even with the Concept Design team helping with the texturing process, most models generated during the first semester did not get textured at all, resulting in levels that were largely gray, texture-less, and full of placeholder models. For the second semester of production,
texturing was much more heavily emphasized. Whereas the philosophy during semester one was to model as many assets as possible and texture them later, semester two assets were required to be modeled and textured sequentially before Modelers were given additional assignments.

In fact, because most of the assets were not textured during the first semester, many non-functioning UV maps were not discovered until the second semester when they were textured, resulting in frustrating delays in production. Some UV maps were so poorly created that they were completely unusable, and often resulted in remodeling the asset with more conscientious geometry and UV maps. Moreover, due to modeling standards not being heavily enforced during the first semester of *High Noon*, most 3D assets had to be revisited for touchups. These assets were redesigned by modelers who were more familiar with modeling for *Unreal* technology and who understood how to properly create UV maps.

It became clear during the second semester that team education would have to be worked into the Chico State Game Studios curriculum to ensure that all team members were prepared to produce work at the professional levels that Chico State Game Studios sought. Regular modeling and texturing tutorials were administered during this semester to the Modeling team in particular, resulting in better textures, more efficient UV maps, and cleaner geometry. In addition, stricter policies were enforced with the Modeling team, putting tight restrictions on polygon limits, texture resolutions, and quality standards.

Senior modelers spent a large portion of the second semester updating weapon and enemy models, some of which were hastily assembled the previous semester. This
greatly impacted the Animation team, due to that that remodeled assets had to be completely reanimated.

**Animation**

Due to a number of modeling setbacks, most of the work completed during the first semester of *High Noon* had to be completely redone to reflect updates to the models in the second semester. Few changes were therefore made to the animation pipeline, though animations created during the second semester of *High Noon* possessed much more originality and personality due to a rising level of comfort with the technology.

The most significant addition to the Animation team’s tasks was the addition of reloading animations to each of the custom weapons. Unlike the native weapons in *Unreal Tournament 2004*, *High Noon* weapons were originally designed to reload. Though reloading was considered to be a significant contributor to the manner in which the game was played, it was considered secondary to implementing weapons that worked on a fundamental level and was saved for the second semester of production.

A senior Animator was assigned to the animation of *High Noon’s* villain, Boz Novak, in the second semester of production. Novak was particularly difficult to implement due to the volume of animations that had to be completed to ensure he was a visually interesting, functioning final boss. Additionally, being humanoid, extra attention was required to make certain Novak moved like a believable human, and did not possess any jerky, unnatural movements.

**Level Design**

The Level Design team during the second semester of *High Noon* had a one-hundred percent retention rate from the first semester of production, meaning that
minimal training had to be administered and work could resume almost immediately at the semester’s start. Because all former Level Designers returned the second semester, no new Level Designers were assigned to the Level Design team, and the second semester of production became a near-perfect continuation of the semester prior. Most Level Designers chose to redesign their levels, repositioning level landmarks and overhauling quests.

Assets, produced at a significantly faster pace than the previous semester, flowed constantly to the Level Design team for integration. While Level Designers did still experience bottlenecking during the second semester, it was on a much smaller scale, with assets arriving at a much more reliable rate. In addition, the adoption of some native Unreal 2004 3D assets helped to alleviate some of the production strain on the Modeling team, and helped keep the Level Design team productive.

Level Designers were required to document their levels in significantly more detail than in the previous semester, submitting lengthy documents specifying the pre- and post-conditions for every planned quest and the suggested dialogue corresponding to each quest. In addition, all Level Design team members were required to make master asset lists detailing what 3D assets they would need to complete their levels aesthetically. Those asset lists served as a foundation for modeling assignments over the course of the semester. Additionally, the documents specifying level quests were sent to the Programming team for implementation, allowing Programmers to carefully plan each quest’s design in terms of technical and custom code requirements.

The Level Design team was also responsible for the design of the endgame, which was not explicitly designed during the first semester of High Noon. A Level
Designer and Programmer were assigned to create a boss fight featuring Boz Novak, wherein he used a number of special abilities to defeat the player, including summoning other enemies into the fight. Though this should have been addressed during semester one, the resulting boss fight was overall effective and challenging, featuring a fully-functioning Novak who used a range of distinctive techniques to attack the player.

Playtesting was emphasized to a much greater extent in the second semester of *High Noon*. Late in the first semester, it was recognized that the lack of playtesting plans in the production pipeline were harmful to ensuring the most important requirement of a completed game: playability. However, even in the second semester, it was difficult to find a way to test levels that were functionally and aesthetically incomplete. Even in preliminary playtests, playtesters complained that it was difficult to discern what they were supposed to be doing at a given point in time, and that level quests were largely unintuitive. The Director and Level Designers were at a loss; playtesting was administered somewhat regularly, but it was often unclear what elements of the level were being tested, and the majority of the data collected from structured playtests was unhelpful or vague.

Conversely, even the small, disjointed playtests administered during the second semester helped spot technical errors in the levels such as areas that allowed the player to exceed boundaries and fall through the map, meshes with missing collision, or areas where enemies consistently stuck to terrain. While the *High Noon* levels were not completely functional, they were largely technically sound due to the playtesters who were frequently finding and reporting design errors.

**Programming**
Programming in the second semester of *High Noon* focused on refining the previous semester’s code and adding new functionality such as ragdoll physics to enemies and reloading functions to weapons. In addition, the Programming team continued work on artificial intelligence, in-game menus, weapon and enemy code, and the complicated Communication Terminals that delivered game back story and quest information.

In the original *High Noon* design document, weapons were designed to reload when they depleted all of the ammunition in a “clip,” meaning that all weapons tracked the amount of ammunition in the weapon and the amount of ammunition the player carries in their “backpack.” In favor of creating weapons that successfully fired and dealt damage, reloading was not pursued during the first semester of *High Noon* and instead saved for the second semester. Special reloading functionality had to be built into each weapon from scratch, largely changing the manner in which the weapons were coded. This was a significant task that took a large percentage of the Programming team’s resources over the second semester.

*High Noon* enemies, when killed, originally played a death animation that caused them to clip through the level terrain, which greatly detracted from gameplay. In *High Noon*’s second semester, we began to experiment with ragdoll physics on our custom enemies. Instead of an animation playing when an enemy was killed, the *Unreal* physics engine would take over the enemy mesh, causing it to crumple and dynamically collapse onto the terrain. With ragdoll physics implemented, the enemies seemed much more alive, and were even entertaining after they were defeated, as they still reacted to the player’s bullets.
Technical Directors

The unexpected increase in team size between the first and second semesters of *High Noon* caused significant growing pains in the production pipelines used for the project, in particular with the distribution of responsibility across the team’s senior members. Technical Directors in particular had only slightly increased responsibility from their fellow team members, with most of the responsibility falling to me for decisions on team management, assignments, assignment feedback, design decisions, art direction, and technical assistance. As the team grew, the distribution of responsibilities became top-heavy, putting unrealistic strain on me as the Director and causing problems with the flow of assets and assignments.

As a result, Technical Directors during the second semester of *High Noon* had increased responsibility to their teams. They were expected to closely monitor their teams, maintain close communications with all team members, and provide technical support when necessary. However, even in the second semester, most major game decisions, scheduling, and assignments still originated from me.

**Top-Heavy Pipeline Policies and Versioning**

Also among my duties during the second semester were version control and asset management, which was largely an artifact from the previous semester when the much smaller team was generating assets at a slower, more manageable pace. I received and categorized all produced assets for the game, developing packages and sending them to the Level Design team for integration into the game world. As time went on, it became
clear that this task was too cumbersome for a single person, and as a result many assets were lost or improperly categorized.

No version control software or localized servers were used during High Noon’s first and second semesters of production. Instead, files were transferred haphazardly through e-mail and portable media such as flash memory drives and portable hard drives. The results from this system were largely negative: many team members did not adhere to any sort of file naming conventions, leading to massive confusion regarding file versioning. The time lost tracking assets without software assistance was significant, and future Chico State Game Studios projects now use localized server space and version control software such as Subversion to track changes made to assets and code.

Playtesting

While parts of High Noon were playable by the end of the second semester, the game did not reach the Alpha release that the team was working toward. One of the largest contributors to High Noon not reaching Alpha was the lack of planned playtesting procedures in the production pipeline. Though it was recognized during the second semester that playtesting needed to be integrated into future pipelines, it was nearly impossible to modify the manner in which the game was being assembled to accommodate the new demands for playtesting.

Ideally, playtesting should have been among the foremost goals during all three semesters of High Noon’s production. However, High Noon began with such a novice group of game developers that the focus instead was to overcome the many technological hurdles and learn how games were assembled using Unreal 2004 technology. The playability of the game was largely lost in the whirlwind of learning to
produce game content on the fly, and the triumph of minor victories such as correctly importing assets blinded the team’s vision of the final goal.

However, the small amounts of playtesting that were completed during the second semester of *High Noon* were valuable to Chico State Game Studios as a whole, proving that games do not have to be aesthetically or functionally complete for players to interact with them. Instead, games can be compartmentalized, and developers can test each segment individually with various playtest groups. Additionally, it helped us understand the value of directed playtesting, wherein a developer selects a function or group of functions to test and creates an environment in which the player can interact with them freely. Data collected from these playtests can be directly applied to the tested functions, improving the game and the manner in which the player interfaces with it.

**High Noon, Semester Three**

Prior to *High Noon*, Chico State Game Studios projects spanned two semesters: one semester focusing primarily on pre-production and production, and the other on continued production and post production. Therefore, during *High Noon*’s second semester, new game ideas were solicited by the Chico State Game Studios Executive Committee, and *High Noon*’s successor, *D.A.V.I.S.*, was selected for the following semester. As with Chico State Game Studios projects prior to *High Noon*, it was assumed that as the second semester came to a close, so did formal production, and that all additional production would be completed outside of the Chico State Game Studios team. While the game was not technically complete, it did demonstrate an impressive proof of concept, and was an extraordinary portfolio piece for the students.
involved. Following precedent, the Chico State Game Studios team prepared themselves for a new game and new Director.

In an unprecedented series of events, the Chico State Game Studios Executive Committee decided that it was time for Chico State Game Studios to switch engines from *Unreal 2004* to the newer, more powerful *Unreal 3* engine. To accommodate the switch, *High Noon* was brought back for a third semester as a buffer project, allowing the team time to familiarize themselves with the intricacies of the new game engine while working on a familiar game concept. Instead of starting production on *D.A.V.I.S.*, Chico State Game Studios would instead work on porting the *Unreal 2004* version of *High Noon* to the new *Unreal 3* engine. By porting *High Noon*, the game would maintain all of its previously designed functionality and experience a large update in aesthetic quality.

In addition to using *High Noon* as a venue with which to familiarize the team with the newer, more complex engine, it also gave Chico State Game Studios a chance to introduce the new Three-Semester System (see Chapter IV, *Changes Made to Chico State Game Studios Policies, Introduction of the Three-Semester System*). Instead of entering full-fledged production, *D.A.V.I.S.* started with pre-production, working in the background of the new *High Noon* conversion project.

The switch to the Three-Semester System and the new *Unreal 3* engine were extremely beneficial for Chico State Game Studios. By allowing three semesters for a given Chico State Game Studios project, a full semester could be dedicated to preproduction, mitigating many of the issues that *High Noon* and previous projects had with poor planning and design. In addition, the adoption of the *Unreal 3* engine gave
students an opportunity to learn techniques currently in use in the game industry, strengthening their portfolios with relevant technologies and professional-quality work.

Revisiting *High Noon* with the same Director for a third semester also allowed further tightening of the production pipelines with massive updates to reflect the new technology, as can be seen in Figure 6. By the third semester of production, *High Noon* had a tight-knit core of veterans, some of whom had worked for Chico State Game Studios since its first major project, *Bronze Age*. Stressing documentation during the previous semesters ensured a more solid knowledgebase among the team, allowing sub-teams to adopt new related technologies and techniques faster than in previous semesters. Overall, the team was more comfortable with production than they had been in the previous two semesters: the strong core of veterans served as on-site trainers for the new recruits, the team was intimately familiar with the story, characters, and vision for *High Noon*, and there developed a renewed enthusiasm for implementing the game using the newer, more sophisticated technology. *High Noon*’s third semester truly was a culmination of two hard semesters of work and experimentation, applied to a new engine with a seasoned team of developers.

Furthermore, the third semester of *High Noon* forced Chico State Game Studios to expand the scope of the pipelines used for video game productions to encompass a much larger team. By the beginning of the *Unreal 3* implementation of *High Noon*, Chico State Game Studios ballooned to fifty-four students, including the two external Audio and Writing teams. This near fifty percent increase in student enrollment meant that best practices used by Chico State Game Studios had to be concise and clear,
Figure 6. Chico State Game Studios inter-team dependencies.
and was a major motivation for the formalization and documentation of many policies and pipelines that had not been completely implemented during prior semesters.

High Noon Semester Three Structural Changes

With two semesters of experience on the project and a strong vision of what High Noon and Chico State Game Studios should accomplish, a significant number of structural changes were made to team organization and production standards before the beginning of High Noon’s third semester. Many team positions were added to Chico State Game Studios, including the Research Lead and Technical Art team. In addition, new production techniques were explored, such as the successful pairing of a Level Designer to a Programmer for the implementation of complicated in-game quests (see Chapter IV, Changes Made to the Chico State Game Studios Team Structure, Pairing Sub-Teams).

Among the significant changes made to the team structure was the addition of a special committee for the final boss fight with Boz Novak. Composed of me, a senior Level Designer, a senior Animator, and a senior Programmer, “Team Novak” was responsible for the design and implementation of the last level of High Noon, which was completely redesigned in the third semester of High Noon to better suit the whimsical roots of the game. The implementation of “Team Novak” resulted in a much more engaging boss battle than in the Unreal 2004 version of High Noon, complete with custom musical score and three major battle phases.

Though Chico State Game Studios was technically structured for the inclusion of an Assistant Director, the position was not filled during the third semester of High Noon because of a senior-level staff shortage. This was a primary motivator for the
continued rebalancing of responsibility between the Director and team leads. Technical Directors during *High Noon* became true team leads, responsible for carefully monitoring their teams, assisting in the assignment of work, providing detailed feedback to team members, and administering training when necessary. Technical Directors during the third semester of *High Noon* served as the foremost point of administrative contact for most team members, allowing the Director time to participate in the management of external teams and the day-to-day workings of the classroom.

External teams were revisited during the third semester of *High Noon* for the first time since the first semester of production. Audio and Writing teams were formally established, who met with me outside of class on a weekly basis. The Audio team consisted of two composers, one of which served part time as an audio engineer for voice acting and narration, and one sound effects artist, who recorded custom sound effects for all of the custom weapons, enemies, and environments in the game. The Writing team, composed of two creative writers, supplied dialog for all of the non-player characters and the narration for the intro cinematic (see Chapter IV, *Stressing the Interdisciplinary Nature of Video Game Design and Implementation*). The addition of the external teams was enormously successful, bringing an increased level of focus to the characters and back story.

The *High Noon* Semester Three Production Pipeline

By *High Noon’s* third semester, the production pipeline for most aspects of production had been carefully constructed and tested over the previous year of production and was easily translated into the new *Unreal 3* engine. Figure 7 shows the flow of
Figure 7. Enemy and weapon implementation during High Noon’s third semester.
enemy and weapon assets through each of the seven Chico State Game Studios teams, and illustrates the most complicated of all of the asset production pipelines. Game elements produced for *High Noon* generally progressed through Chico State Game Studios teams in this nature: starting with the Concept Design team and ending with the Programming or Level Design teams for integration in-game.

**Concept Design**

The new engine brought a large number of changes to the original *High Noon* concepts. To reflect the capabilities of *Unreal 3*, all of the weapons and enemies for *High Noon* were redesigned to reflect the engine’s increased detail capabilities. In fact, most of the characters and art generated for *High Noon* received a facelift with the switch to the new engine. All of the game’s non-player characters, Jekyll, his alter-ego Hyde, and Boz Novak all experienced significant character redesigns during *High Noon*’s third semester. The new characters were more sophisticated than their *Unreal 2004* counterparts, bringing new dimension to the aesthetics of *High Noon*. In addition to all of the characters, weapons, and enemies, the user interface experienced a significant redesign to reflect the more sophisticated menus of which *Unreal 3* was capable.

Once reconcepting was finished for all of the major *High Noon* assets, Concept Designers completed a range of high-quality concepts, including further visualization of the game’s background and mythology. An interesting side effect of the continued refinement of the game world surrounding *High Noon* was the creation of fan art and the generation of elaborate character back stories. Aspects of the *High Noon* game world that were never explored in either the original design document or the game itself were extensively developed by the Concept Design team, resulting in a cult following of
many of the characters, in particular Boz Novak. The fan art and back story production had an unexpected effect on the team’s morale, revitalizing interest in the project and unifying design decisions. Suddenly, the majority of the team knew many parts of the complicated *High Noon* back story instinctively, and the *High Noon* world widely felt like it had taken on a life of its own.

**Modeling**

Like the semester before it, the third semester of *High Noon* brought tighter production pipelines and higher standards for modeling and texturing. With the use of the *Unreal 3* engine, Modelers had significantly more geometry at their disposal to create their 3D assets. Whereas in *Unreal 2004* Modelers faced polygon restrictions that resulted in blocky, low-resolution models, *Unreal 3* could handle well over double what its predecessor engine could process. *High Noon* assets became much more geometrically complex, resulting in richer and more immersive game environments.

First Tier assets had even higher polygon allotments, with most weapons and enemies approaching five thousand triangles in their redesigned state, compared to the eight hundred to fifteen hundred polygon limits of the *Unreal 2004* versions. The resulting models were significantly more visually appealing, allowing for the use of more sophisticated technologies such as normal mapping and emissive textures. See Figure 8 for an example of this effect as demonstrated by the *Unreal 2004* version of the Red Shirt’s Revenge Ray Gun and its *Unreal 3* counterpart.

Most Second Tier assets created for the *Unreal 2004* version of *High Noon* were either significantly altered to fit into the new artistic direction that occurred with the engine switch, or thrown out altogether and redesigned by the new Modeling team. While
Figure 8. The Unreal 2004 Red Shirt’s Revenge Ray Gun versus the Unreal 3 Red Shirt’s Revenge Ray Gun. Unreal 2004 model and texture by Andrew Holifield; Unreal 3 model by Scott Washington; Unreal 3 texture by Alisha Thayer.

the assets that were created during the first two semesters of High Noon were certainly production quality for Unreal 2004, when they were placed in Unreal 3 they looked unfinished and out of place.

In short, almost none of the 3D assets created for the Unreal 2004 version of High Noon were of a sufficient quality for the new Unreal 3 engine and had to be updated or completely redesigned before they were ready to be used in the Unreal 3 version of High Noon.

Animation

Recreating all of the models for High Noon had serious repercussions for the Animation team, who were tasked with re-rigging and reanimating all of the original High Noon assets for Unreal 3. Because of the nature of rigging and animating, no
original animations from the *Unreal 2004* version of *High Noon* survived the move to *Unreal 3*.

Like during the first semester of *High Noon*, the Animation team suffered bottlenecking during the third semester due to the fact that models had to be finalized before they could be sent to be animated. With nothing to animate for the first half of the semester, the Animation team joined the modeling team and assisted in the creation of First and Second Tier Assets. Many animators animated their own models, increasing their efficiency because of their familiarity with the mesh’s geometry, which proved to be an effective means of ensuring that all team resources were being used at all times.

Animation also felt their biggest crunch during the third semester of *High Noon*; because of the time they had to wait for assets to make it through the pipeline, they were effectively allotted one-half of a semester to animate all of the moving weapons, enemies, and environmental elements needed to complete the game. Though the work was completed on-time, it required an enormous push of frantic work to ensure all of the animations were remade according to specifications. In the end, though the animations created for the *Unreal 3* version of *High Noon* were of very high quality, some of them contained errors or unpolished elements that did not exist in their *Unreal 2004* counterparts due to trying to achieve one year of work in a few weeks.

**Level Design**

Like many teams with returning members, the Level Design team was excited to have an opportunity to revisit their levels and implement more refined terrain, object placement, level objectives, and flow. Because of the new scripting capabilities of the *Unreal 3* engine, Level Designers were able to plan for more sophisticated triggered
events and quests. *High Noon* levels using *Unreal 3* technology were technologically stronger, allowing for easy implementation of game elements that took weeks of experimental scripting to achieve in *Unreal 2004*. The visual scripting editor, Kismet, allowed Level Designers to create scripts for their own simple level features like swinging doors and flickering lights.

Significant tightening to team training and the Level Design pipeline occurred during the third semester of *High Noon*. Because of the switch to a new level and game editor, the Level Design team underwent “Level Design Boot Camp,” wherein they were instructed to view video tutorials created by Epic Games, read articles about the important functions and changes to *Unreal Editor*, and assemble some simple levels to demonstrate mastery of the topics necessary to create levels using *Unreal 3*. As a capstone, each Level Designer was administered a verbal test (see Appendix F) wherein they demonstrated important skills to the Director to verify a sufficient level of comfort with the technology. This was added to *High Noon*’s Level Design policies because of difficulties Level Designers were having with *Unreal Editor* during the first two semesters of *High Noon*’s production, which was greatly attributed to lack of formalized training.

It was expected when the decision was made to move *High Noon* to *Unreal 3* that levels would be exported from *Unreal 2004* and simply imported into *Unreal 3*, allowing the Level Designers to keep the preponderance of their work from previous semesters and using it as a foundation for the new *Unreal 3* levels. However, when it came time for production, it was discovered that the importing software was not compatible with *High Noon*’s large and complicated levels. Like the Animation team,
Level Design had to start completely over on the design of their levels, rebuilding them from scratch in the new *Unreal Editor*.

Like in previous semesters, the Level Designers began by paper prototyping their levels on grid paper and ensuring that written treatments were prepared and sent to the Director if any changes occurred from the previous semester’s level incarnations. Once the prototypes were completed, BSP, or *binary space partitioning*, representations of the levels were constructed in *Unreal Editor* to check for scale and basic level flow. Like in *High Noon*’s first and second semesters, the Level Designers faced serious pipeline bottlenecking while they waited for assets to be completed by the Modeling team.

Pipeline stalls during the first two semesters of *High Noon* increased awareness of how damaging waiting for assets can be to a project. To mitigate this, a new method of production was introduced into the Level Design pipeline. This method, called “Designing by Companion Cube” (see Chapter IV, *New Chico State Game Studios Design Strategies, The “Companion Cube” Approach to Level Design*) allowed Level Designers to create representations of their final levels by using one cube-shaped mesh as a stand-in for all planned in-game meshes. Resourceful Level Designers constructed complicated mesh stand-ins such as trees and buildings by creatively distorting and combining multiple iterations of the “Companion Cube” mesh. By using a cube to represent all of the placed meshes in a given level, Level Designers were no longer dependent on the Modeling team, thereby eliminating nearly all of the Modeling-Level Design bottlenecking.
As meshes were integrated into the game, Level Designers replaced the “Companion Cube” stand-ins with finalized High Noon meshes. These allowed them time to iteratively design and carefully plan the manner in which meshes were lit in-game. The use of the temporary meshes in levels helped playtesting occur much earlier in the pipeline than in previous semesters of High Noon and greatly increased the quality of levels overall. In addition, Programmers were able to work early on with Level Designers to assist with the implementation of vital quests and triggers well before the levels were aesthetically complete. Dedicating Programmers to Level Designers ensured a constant development of level functionality from simple level behaviors such as opening doors to more complex level elements like triggering explosions and enemies.

The changes made to the Level Design team were significant during the third semester of High Noon; not only did Level Designers have a firmer grasp of the technology they were using to create High Noon levels, but they were able to implement their ideas quickly, ensuring time to test and review how effective the designs were and how they effected the game as a whole. Though the Level Design team still felt the effects of asset-related bottlenecking, it did not cripple the team as it had in the past, and allowed for a level of rapid production previously unattained by Chico State Game Studios.

Programming

Like Level Design, the Programming team expected to transfer most of the existing code from the Unreal 2004 version of High Noon over to the new Unreal 3 engine. However, like with Level Design, Programmers were surprised to find most of the original High Noon code to be incompatible with the newer version of UnrealScript
that *Unreal 3* employed. The first direct conversion of *High Noon* code resulted in the compiler throwing ninety-two errors, which resulted in the discovery of more errors upon the resolution of the old ones. "After playing with it a couple of times, we basically decided that we were going to have to just start from scratch" (C. Funk, Chico State Game Studios, personal communication, April 2, 2009).

The implementation of all of the weapons, enemies, and artificial intelligence from scratch was a frightening endeavor for the ten person Programming team. In addition to rewriting the core gameplay, the in-game menus, head-up display (HUD), communication terminals, and custom item pickups had to all be completely rewritten and re-implemented. Not only did the Programming team have to quickly learn how to program using new *UnrealScript*, but they also had to learn the new *Unreal 3* file hierarchies, research new methods of implementation, and re-implement over a year of work in thirteen weeks.

In addition to the responsibility of re-implementing all of the game’s prior functionality, many programming approaches required massive redesigns because of the heavy use of *Matinee* and *Kismet* in *Unreal 3*, which did not exist in *Unreal 2004*. The *High Noon Unreal 2004* HUD, for example, used many functions that simply did not exist in *Unreal 3*, and took one Programmer nearly half of a semester to redesign and re-implement using new methods. In addition, quests and triggers controlled by a number of custom classes and initialization (INI) files in the *Unreal 2004* version of *High Noon* were completely redesigned to use *Kismet*, *Unreal’s* built-in visual scripting editor.

Versioning assets proved to be a significant issue for most teams during the first two semesters of *High Noon* because the project used no central server or version
control software. Instead, files were haphazardly transferred using portable media and e-mail, which resulted in conflicting assets and lost work. During High Noon’s third semester, a server was built and housed in the College of Engineering, Computer Science, and Construction Management, and was used as a check-in/check-out system for all code-based assets. Due to limited server space and bandwidth, art assets were not housed on the High Noon server.

Under High Noon’s first and second semester pipelines that lacked version control, the Programming team would likely need an additional year to complete enough work to make the Unreal 3 version of the game functional on an Alpha level. However, implementing a server running Trac and Subversion unified the Programming team, allowing them to quickly and accurately create, update, and track builds of the game. Code that broke any element of the game could be easily reverted, and code was never lost in the shuffle of files. Rather than having to contact me for an updated build of the game, Programmers could instead download it directly from the server’s repository.

Version control software, in combination with the documentation and experience accumulated from previous semesters, allowed the Programming team to work quickly; generating code at roughly twice the speed of previous projects.

Audio

The Audio team was one of the three new additions to the High Noon team structure. Composed of two volunteer part-time musicians and one volunteer part-time sound effects engineer, the Audio team met outside of the weekly High Noon classes with me to develop the aural identity of the project.
The Audio team was responsible for writing custom music for each of High Noon’s four levels, special music for the new elaborate boss battle, and supplying new music for the introduction cinematic, in addition to creating sound effects for all of the weapons, enemies, and special environment elements. Each of High Noon’s levels had an “idle” and “combat” state that required distinctly identifiable musical score that could seamlessly blend from one to the other as the player explored the levels and encountered enemies. In all, twelve musical tracks and over one-hundred sound effects were generated during the third semester of High Noon’s production.

In addition to composing music and designing sound effects, the Audio team was also responsible for recording all of the in-game non-player character (NPC) dialogue and designing post-processing techniques so that the resulting sound bites successfully integrated into the High Noon world.

Writing

Prior to the third semester of High Noon, all writing for dialogue and cinematics occurred in-house, and was often authored by the Director. With the move to the new engine, two part-time Writers were brought onto the team to revisit old written materials and write new dialogue. Like the Audio team, the Writing team met externally once a week with me and reported directly to me with progress. Furthermore, the Writing team assisted in the further development of the three non-player characters, Boz Novak, and tightening the existing High Noon plot.

The Writing team’s first task was to revisit the game’s opening cinematic and rewrite the old narration, which was darker and gloomier than was appropriate for the image that High Noon was trying to establish. By writing new opening narration and
advising on the game’s story, the Writers unified the *High Noon* story and characters, bringing remarkable cohesion to the game overall.

Unlike the Audio team, the Writing team functioned well as a part-time, external team. Remaining in contact with the Director via e-mail and weekly meetings was sufficient for producing high-quality, relevant content. Revisions to scripts and narrations occurred iteratively between the Writing team and the Director, ensuring that changes made to the plot and characters remained consistent with the existing plot. The positive results from the addition of the Writing team were clear by the end of *High Noon*’s third semester, with a strong cast of unique characters and a concrete game identity.

**Technical Art**

The Technical Art team was established late in *High Noon*’s third semester to account for the new, “next-gen” demands of the 3D asset pipeline. Originally created as a sub-set of the Modeling team, it was the responsibility of the Technical Art team to research the proper use of the new material editor in *Unreal 3* and develop standard practices for importing textures and creating various in-game surfaces. Instead of passing their completed 3D models to me, Modelers would instead give their completed models and diffuse textures to a Technical Artist, who would then generate normal and specularity maps based off of the submitted diffuse texture. The Technical Artist would then import the three textures (diffuse, normal, and specularity) and assign them via *Unreal*’s node-based surface editor. This complicated process requires a working knowledge of node inputs and outputs, as well as various material channels and node
types. All models in *High Noon* were passed through the Technical Art team before being integrated with the game.

Technical Artists were also responsible for designing special textures that required emissive, transparent, reflective, or parallax elements. Models with surfaces such as chrome or glowing goo were sent to Technical Art with special specifications and required significant experimentation on the part of the Technical Artist to create a surface that was convincing and fit with the existing art style of the game.

In addition, Technical Artists were tasked with assembling asset packages and passing them to the Level Design team for placement in-game. Technical Artists adhered to strict naming conventions to ensure that Level Designers could quickly locate assets and asset updates. *High Noon* naming conventions for assets followed the rule: 

`LevelName_AssetType_AssetName_VariationNumber`. For example, all assets belonging to the Sutter’s Mill level had the prefix “SM_”, props contained the identifier “Prop”, a barrel would be called, “Barrel,” and, if there were many variants of barrels, it would be noted using a number. Therefore, for a barrel native to Sutter’s Mill that is the third variation would be called “SM_Prop_Barrel_3” and be placed in a package exclusive to the Sutter’s Mill level.

Though Technical Art started as a modeling sub-set, it quickly evolved into its own separate team due to the differences in production pipelines. A Technical Director and Research Lead were established to reflect the change into a major sub-team. The Technical Art team is unique, however, because a Chico State Game Studios project does not generally need a Technical Art team until 3D assets have begun to reach the end of the production pipeline. Therefore, it is a wise practice to keep all team members
interested in participating in Technical Art on the Modeling team until the team is ready to form a Technical Art team and start training.

*High Noon* Semester Three Production
Pipeline Observations

Unlike in previous semesters of *High Noon*, the third semester of production concluded with a solid feeling of accomplishment and soaring morale. Pipelines worked more efficiently than ever before, all teams were more skilled and knowledgeable than on any previous Chico State Game Studios team, and there existed a unified vision and drive for completing the game. The biggest limiting factor on *High Noon*’s third semester was time, with an unrealistic expectation of completing a year of work in a new, more complicated engine in thirteen weeks. Production during *High Noon*’s third semester ran at a consistently accelerated pace, and was effectively a thirteen week crunch to complete the game.

**Asset Management**

Despite the overwhelming success of most of the pipelines, the lack of version control on the art assets in *High Noon* was a serious cause of bottlenecking and lost work, as in previous semesters. Had Chico State Game Studios access to the server space to store the nearly 40 gigabytes of models and textures created for the game, it is likely that art assets could have been generated at a more rapid pace and with fewer errors. Moreover, tracking the progress of assets would have been considerably easier with a central server housing all of the team’s work rather than the haphazard flash drive and e-mail methods used throughout *High Noon*’s production.
In a related issue, the file sizes involved in creating a “next generation” game were significantly greater than those of the previous generation. The final build of *High Noon* using the *Unreal 2004* engine was 300 megabytes, whereas the *Unreal 3* version of *High Noon* topped out at almost 2.5 gigabytes. This affected distribution options (2.5 gigabytes is far less accessible than 300 megabytes in terms of download times) due to the fact that *High Noon* heavily relied on electronic distribution. Not only that, but file management because a significant issue with the move to *Unreal 3*. Not only did storage capacity on the team need to significantly increase, but larger files meant longer load and import times. Particularly with the Technical Art team, Photoshop documents (PSD’s) could take upwards of three minutes each to load into *Unreal Editor*. Considering that each texture yielded between two and four additional PSD’s of similar size, a model could require the importation of at most five individual PSD’s and cause fifteen minutes of wait time per model. This proved to cause serious delays, especially if images had to be re-imported to fix errors or reflect updates.

**The Formal Addition of the Audio Team**

The addition of the Audio team was a significant breakthrough for Chico State Game Studios, allowing for the creation of more professional, sophisticated games. Though musicians were employed for the generation of audio assets during the duration of *High Noon*’s third semester, the responsibility of integrating those assets could not be delegated to the part-time volunteer staff and was therefore my responsibility. Like in previous semesters of *High Noon*, delegating such as massive task to the already-busy Director, in combination with the fact that the game was being completely rebuilt from the ground up, resulted in massive delays in the integration of audio. By the end of the
semester, only a fraction of the audio created for *High Noon* had been successfully integrated into the project.

The addition of the Audio team was a significant success for *High Noon*. The quality of audio assets produced was unprecedented for a Chico State Game Studios project and added significant depth to the game’s identity and effectiveness. Because of the difficulties that occurred with integration, it became apparent that in future Chico State Game Studios productions, the Audio team would need to be integrated full-time into the organization to ensure that assets were quickly and accurately implemented in-game.

**The Consequences of Porting**

The idea of “porting” the game was an attractive one when the Chico State Game Studios Executive Committee was considering the move to *Unreal 3*. Considering that most of *High Noon* had been completed by the end of the second semester of production, it was widely thought that most of the semester could be spent learning the new technology and polishing the existing assets. However, technological incompatibilities with the software that was going to be used to port *High Noon* resulted in having to completely remake the game using the new technology. The goal of learning the new *Unreal 3* software, redesigning all of the code assets, recreating all of the art assets, and redesigning all of the levels for *High Noon* in thirteen weeks was unrealistic even with a team of fifty-four. While *High Noon* proved to be an excellent transitional project and was integral in preparing the team for production on *D.A.V.I.S.*, the prospect of finishing off the four levels and custom boss battle by the end of the semester was lofty at best. *High Noon*’s experience with new technology should serve as a warning to
future projects: though new technology is certainly attractive, it almost always brings
with it hidden technological difficulties, lack of documentation, and unforeseeable
production bottlenecks. When working with new technology on a project, it is important
to plan for the potential difficulties that can result from using unfamiliar software and
production techniques.

*D.A.V.I.S.*, Semester One: Preproduction

The preproduction semester of *D.A.V.I.S.* ran concurrently to the third semester of *High Noon* and was largely treated as an opportunity to test pipelines and theories developed during *High Noon* that could not be tested due to time constraints or lack of project flexibility. It was also a valuable chance to verify that processes developed during *High Noon* indeed worked with a project of a completely different nature: whereas *High Noon* was an action first-person shooter laden with guns and whimsical enemies, *D.A.V.I.S.* was designed to be a psychological thriller with no enemies, driven primarily by environmental storytelling and character interaction. The opportunity to test *High Noon*’s pipelines on a game of a different genre not only helped validate the existing pipelines, but helped continue to refine Chico State Game Studios practices and team structures for future projects.

The foremost goal of *D.A.V.I.S.*’s preproduction semester was to establish preproduction techniques that had been up to that point only theoretical. Preproduction on *High Noon* was completed before the implementation of the Three-Semester System (see Chapter IV, *Changes Made to Chico State Game Studios Policies, Introduction of the Three-Semester System*) and was therefore performed, according to precedent, by the
Director over the summer preceding the fall production semester. As a result, the preproduction for *High Noon* was weak at best, containing many errors and inconsistencies, as well as an unacceptable number of underdeveloped game elements.

Instead of the design of the game being the sole responsibility of the Director, a Preproduction team was formed the summer before preproduction began to establish the game’s plot, game play, characters, overarching game goals, use of technology, user interface and game systems, and all other information needed to make a successful game. By establishing a Preproduction team, the Director could draw upon the expertise of many people from different technological and disciplinary backgrounds. Therefore, instead of the Director designing in a creative vacuum over the summer before the start of the project, he or she could debate creative decisions with knowledgeable Chico State Game Studios veterans and verify that designs were not only feasible, but consistent with other decisions made for the project. The quality of design during preproduction on *D.A.V.I.S.* was of significantly higher quality than that of *High Noon*.

The Preproduction team (see Chapter IV, *Changes Made to Chico State Game Studios Policies, Introduction of the Three-Semester System*) for *D.A.V.I.S.* was composed of the Director, Assistant Director, an Art Director, a senior Programmer, and later a Level Designer. I served on the committee as the Art Director, bringing with me my experience as Director of *High Noon*. This proved to be very helpful as there were many issues that had occurred during *High Noon* that were relevant to decisions made in the design of *D.A.V.I.S.*

The Preproduction team began to meet informally over the summer before the preproduction semester. During this time, early ideas for the *D.A.V.I.S.* game were
established, as well as a tentative schedule for the goals and milestones of the preproduction semester. Once the preproduction semester started, the Preproduction team met weekly for between three and six hours to discuss and brainstorm the complex story behind *D.A.V.I.S.* and the goals for production.

Because *D.A.V.I.S.* was designed by the Preproduction team, it naturally metamorphosed from the original, *Resident Evil 4* inspired game that was pitched to Chico State Game Studios to a much more cerebral, psychological thriller that harkened to the adventure games of the 1990s. *D.A.V.I.S.* was iteratively designed, with each creative meeting resulting in massive changes to the game’s format until a solid story was developed about midway through the preproduction semester. In fact, early versions of *D.A.V.I.S.* featured warring factions, meteors, and science fiction elements that were quickly phased out of the game in favor of more psychological thriller elements.

A significant and positive change that came with the addition of the preproduction semester and the Preproduction team was the integration of paper prototyping into Chico State Game Studios’ standard production practices. Prior to *High Noon*, games were green-lit for production well before they demonstrated any sort of playability, with the only production requirement being a working design document and appointed Director. The addition of paper prototyping ensured during preproduction that game elements were being checked for functionality and fun before the game went into production, which helped prevent late-production changes. Whereas *High Noon* was designed without consideration for how playtesting fit into the already tight production pipeline, *D.A.V.I.S.* was designed to include frequent, iterative playtests at all major phases of production.
One of the most significant successes of the Preproduction team was catching and eliminating game elements that were either too difficult to design for or that conflicted with other game objectives before the start of production. The longer a game is in production, the harder it is to make changes to the game’s design and the more costly changes become. *High Noon* experienced this to a large degree when it was discovered late in the third semester that one of the game’s core features, the relationship between Jekyll and H.Y.D.E. (see Chapter IV, *Cautions for New Directors*, *The H.Y.D.E. Dilemma*), was unbalanced and greatly detracted from the game’s playability and fun. At the time this was caught, *High Noon* was too far into production to change the Jekyll/H.Y.D.E. relationship and could do little to modify the game without causing irreparable damage to the existing assets and code. *D.A.V.I.S.* encountered a similar situation during preproduction with one of its early core game functions: a remote that the player finds halfway through the game that provides them with a means of solving puzzles and interacting with the *D.A.V.I.S.* world. The Preproduction team worked for weeks under the assumption that the remote was a necessary element of *D.A.V.I.S.*’s design and spent countless hours trying to make puzzles that were interesting and challenging. It was eventually agreed upon that the remote was limiting the design of the game and was ultimately cut. Cutting the remote during preproduction was relatively simple, requiring only a few design and documentation changes; whereas if the decision had been made to cut it later in production, it may have caused crippling delays in production or unreasonable demand for redesigned assets.

Overall, the addition of the preproduction semester and the Preproduction team was a noteworthy success for Chico State Game Studios by increasing the quality of
the games designed for production. Not only that, but by implementing stronger preproduction, Chico State Game Studios now functions more like studios in industry than in previous semesters.

Results

The Chico State Game Studios Pipeline: A Walkthrough

High Noon’s three semesters of production and D.A.V.I.S.’s preproduction semester were a successful experiment in iteratively developing production pipelines customized for a student game studio with one foot in industry and the other in academia. Chico State Game Studios runs more smoothly and efficiently than it ever has, and is successfully generating skilled production members who possess valuable experience working in large teams that directly translates into the video game industry. This section will break down the results of this pipeline experiment into concrete semesters, detailing the tasks that should be completed to ensure a successful production.

Incoming Chico State Game Studios Directors should find this section valuable in planning their own projects and identifying potential bottlenecks or other project-endangering factors. Chapter IV will supplement the resulting “best practice” pipelines from the year and a half of production covered in this chapter.

The three-semester pipeline is designed specifically for Chico State Game Studios productions using the Unreal 3 game engine with team sizes between forty and fifty people. Specifically, this pipeline was designed and tailored for use during the High Noon and D.A.V.I.S. projects. It should be noted that while a majority of the recommended techniques and practices will fit well into future Chico State Game Studios
projects, production pipelines are expected to continually evolve to suit the unique needs of each game project as they arise. As such, these pipelines should be used as a framework in which to design future game projects and to familiarize incoming Chico State Game Studios students with the manner in which complex, large-scale game projects are produced.

All Chico State Game Studios projects work in a three-semester cycle, with current and incoming projects overlapping one semester to facilitate a regular stream of tasks for the majority of the production team (see Figure 9). New projects are evaluated in the spring from a pool of student-proposed game ideas. Upon selection of an idea, a

![Figure 9. Production cycles illustrated with Chico State Game Studios’ three-semester system.](image-url)
Director is appointed to design, organize, and execute the game project from start to finish over the course of three academic semesters. The Director is charged with forming all major teams to complete the project. The first is the Preproduction team, established during summer, and before the formal preproduction semester of the project. In the following fall semester, the Director and the appointed Preproduction team meet weekly to design all aspects of the new game, documenting their work in the initial game design document. This document is then used in the following semesters of production, when the Chico State Game Studios team expands from the initial six Preproduction team members to a production size of around fifty, divided up over seven major sub-teams of specialists. Production continues for two semesters until a Beta version is produced and released to the public in December.

**Game and Director Selection**

New Chico State Game Studios projects are selected during the spring semester while the current game is in its first semester of full-fledged production. Game ideas are solicited from students, who then voluntarily author game concept documents and submit them to the Chico State Game Studios Executive Committee for review. Game concept documents usually range from two to four pages, and essentially present a high-level game concept. High-level game concepts include information on game genre, target audience, proposed platform, back story, gameplay breakdowns, and a short competitive analysis. In the last weeks of the semester, these game concept documents are shared by their authors at a public presentation. At this presentation, the Chico State Game Studios Executive Committee evaluates the game ideas for originality and
feasibility, later debating their decisions at an external meeting wherein the game is selected.

As part of this process, the Executive Committee searches for an undergraduate student with at least one-and-a-half academic years remaining in their degree program who will serve as the Director of the upcoming project. This student is usually of senior class standing, a Chico State Game Studios veteran, familiar with the processes and technologies that the organization uses, and is comfortable managing large numbers of creative people on long-term projects. The Director stays with the project for three semesters, and may begin preproduction preparations as soon as the new game and Director are made public.

On conclusion of the project, the game is released and presented. Often, the new Director takes five minutes at the end of the Semester Release to provide a sneak peek of the new project to the Chico State Game Studios team and general public. Like the game concept document, this presentation generally covers the high-level concept of the game and is meant to increase awareness and generate enthusiasm for the new game.

Preproduction

As illustrated in Figure 4.3, preproduction for an incoming game project occurs during the last semester of production on the project it follows. Preproduction largely takes place outside of the main Chico State Game Studios team, and is completed primarily by a small creative subsidiary. This subsidiary, referred to as the Preproduction team, meets weekly outside of the primary Chico State Game Studios activities and lectures for three to five hours per week, or whatever is appropriate for the genre and complexity of the project.
The Preproduction team should be formed over the summer before the preproduction semester of the game project. Optionally, preproduction may begin during this time, but is not required due to the difficulties that typically arise with trying to organize students outside of a major academic term. The Director should use this summer in a preparatory fashion: establishing the direction of the game based upon the student-authored high-level concept supplied by the Chico State Game Studios Executive Committee. During this time, the Director should also devise a schedule for the Preproduction team. This primarily consists of identifying what tasks and milestones must be met by the end of the preproduction semester. The Director is the primary motivational force behind the preproduction semester.

The more disciplines represented on the Preproduction team, the more balanced the generated ideas and decisions will be. For example, an ideal Preproduction team might be composed of the current and former Directors, the current Assistant Director, at least one senior programmer, a game designer, and an experienced artist. Designing with a well-rounded team decreases the chance of such oversight as major game design errors slipping into production, whereupon it might be significantly more difficult, or even impossible, to fix them when they are later discovered (see Chapter IV, Cautions for New Directors, The H.Y.D.E. Dilemma).

Creative preproduction meetings are led and organized by the Director, who is responsible for preserving the creative vision of the project throughout production. As these meetings are often structured as large brainstorms, with many ideas being discussed, detailed notes should be taken and distributed to the Preproduction team before the next meeting. These notes serve two major purposes: to ensure that all team
members are aware of current iterative design decisions, future tasks to complete, and overall direction of the project; and to track the progress and evolution of the project over the course of the semester.

Throughout the preproduction semester, the Director also begins to assemble the formal treatment of the game in the form of a design document. The design document acts as a cumulative description of all elements necessary to the completion of the game. Creation of this document is a significant undertaking, due to its large size and level of complexity. It is the responsibility of the Director to finish this document by the end of the preproduction semester. The design document should contain specifications for:

- Game background and back story (including tone and relevant contextual information)
  - Description of player-character and non-player characters (if applicable)
- Target audience
- Intended platforms (e.g. PC, console, et cetera)
- System requirements
- Detailed gameplay description
  - Controls
  - Explanation of necessary interfaces
  - Scoring and win/loss conditions
- Game features
  - Weapons (if applicable)
  - Enemies (if applicable)
  - Important game items (inventory items, et cetera)
- Detailed level/environment descriptions
- Detailed quest/game objective descriptions
- Major development tasks including required technical research
- Estimated schedule

While it is understood that this document will change as the game goes into production, it should be an accurate representation of the game in its current form as it approaches full-production.

Once the game design document nears completion, the Director can use it to recruit the necessary talent to fill any gaps in production staff. It is rare, for example, to find a musician in APCG 495: Advanced Production, necessitating recruitment of such talent from outside of the project. Likewise, writers, video specialists, and voice actors are commonly sought after in the same way. The use of the game design document in recruiting new talent allows the Director to quickly and succinctly communicate the demands of the project, as well as give potential talent a preview of what the game is expected to look like upon completion.

Concurrent with the creation of the game design document, paper prototyping should occur frequently as the preproduction semester comes to a close. The purpose of this is to verify that the ideas generated by the Preproduction team will function as an entertaining, effective game. The members of the Preproduction team are ideally familiar with popular methods for paper prototyping, and are able to select one that adequately tests the core gameplay elements. Playtesting can and should extend through the first semester of production, involving as many team members as possible in the playtesting process. A wider playtest audience will catch more errors earlier, before the game is so
far into production that changes become too intensive to implement. As part of the Director’s authorship responsibilities, any game design changed resulting from playtesting are integrated into the design document.

At the end of the preproduction semester, the Director presents the new, completed game idea to the public at a Chico State Game Studios sponsored release of the preceding game, often taking the last thirty minutes or so to showcase the new project. The completed design document is then distributed to interested parties, and preproduction formally comes to a close as the team shifts focus from the released game to the new game.

Production

Spring marks the first major semester of production for a Chico State Game Studios game project. Teams are formed, tasks are defined, pipelines applied, artistic direction established, and the foundation of the working electronic game is created. This is an exceptionally fragile time for a video game project as it has little established identity, no production standards, and many unknown variables such as team sizes and skill levels.

With that in mind, a Chico State Game Studios project must be flexible in response to the voluntary nature of the organization. Up to this point, the Director can only estimate the number of people in the project and in any given sub-team. Since the Director can’t foresee significant staff shortages until the end of the first night of class, he or she must be prepared to both modify the game quickly to suit the team and to recruit additional talent if any sub-team is critically short on members.
The first night of a Chico State Game Studios project is crucial for communicating the high expectations for the project. This includes presenting the academic workings of the class, reviewing the course syllabus (see Appendix D) and clearly explaining the expectations for each participant. It is wise to stress the difficulty of the course during the first night of class, allowing students an opportunity to carefully weigh their schedules before committing to Chico State Game Studios for the semester. Students must understand from the first night that Chico State Game Studios maintains a policy of mandatory attendance, and that frequently missed classes will greatly affect letter grades. Additionally, it should be clear that Chico State Game Studios relies upon strict, cascading deadlines, and therefore late work cannot be accepted. Students should be instructed to join the class Google Group (see Chapter IV, Changes Made to Chico State Game Studios Policies, Mandatory and Encouraged Methods of Communication) to receive official class e-mails and gain access to important files. On the Google Group, they should download the game design document and read it during the first week of class to ensure that they are knowledgeable about the game they are creating.

In addition to project requirements, the structure of the Chico State Game Studios organization should be covered, explaining all team member roles and responsibilities within the project. Production pipelines should be receive at least a brief explanation, providing interested students with some understanding of the complex and sometimes delicate process used in making video games. Communication both on a local, sub-team level and a global, team-wide level should be heavily stressed during the first night. Chico State Game Studios must work as a unified organization pushing toward the same goal. Without good communication, the project can suffer great setbacks, damaging
the fidelity of the game to the original idea and even hampering completion of the project. Additionally, students need to understand that, while Chico State Game Studios is a heavily structured organization, it is by no means a bureaucracy. The Director should be equally accessible to all team members, and all team members are encouraged to talk to anyone on the team regarding creative ideas, help with technical difficulties, or any other issues that arise at any time during the project. Furthermore, it should be clearly explained that no creative process happens in a bubble; performing as an isolated unit on the Chico State Game Studios team is detrimental to the project and is therefore unacceptable.

The Director should quickly administer a number of short, informal surveys during the first night: getting a brief show of hands for number of return Chico State Game Studios students versus new students; asking students about which sub-team they wish to join; and surveying the class for parties interested in Technical Director and Research Lead positions. This will allow the Director to immediately form a preliminary picture of the new Chico State Game Studios team, where staff shortages exist, and the amount of training that will have to be administered to the team before they are ready for production.

The Director should then collect documents from each team member detailing the contact information they feel comfortable providing to the class, including: their names, e-mail addresses, phone numbers, and instant messaging (IM) information. Also included in the document should be the team they desire to work on and if they are interested in the Technical Director or Research Lead positions. The Director and/or Assistant Director then synthesize this information into a document that is released to the
class to provide instant access to team contact information and detailing the breakdown of teams.

Upon completion of these initial administrative tasks, the Director is then free to explain the creative plan for the semester, including an in-depth description of the game project and an overview of its implementation. The Director should cover the game project in as much detail as possible, setting up the story, characters, gameplay, et cetera, to promote a clear understanding of the project. He or she should welcome questions, making the presentation largely a dialogue to ensure the project is being represented clearly.

By the end of the first week, the Director and Assistant Director should have the Team Contact Information document compiled and available in PDF format on the Google Group. Additionally, the Assistant Director should spend the first week of class finding server space for the project (a minimum of 200 GB for Unreal 3-based projects), setting up and testing version control software, and establishing a skeleton for the internal wiki that the project will use for documenting important specifications for the project.

The internal wiki is a new tool being used by Chico State Game Studios to help mitigate past issues with keeping the game design document updated and relevant, which impacts the Director’s ability to keep team members informed about major changes to the game’s style or design (see Chapter IV, The Problem with Design Documents). By using wiki technology, all team leads can modify the project’s internal wiki to reflect changes made in their area of the project, alleviating the burden on the Director and allowing design changes to be documented in real-time. In addition, wiki
versioning features allow readers of the document to easily track changes and navigate to pertinent new information instead of scrubbing pages, manually looking for changes.

By the start of the second week the Director and new Technical Directors should have Assessment Assignments (see Appendix E) organized for all of the sub-teams for distribution at the beginning of the week. Assessments will vary depending on the sub-team, but should all be designed to evaluate the current skill level of each team member. Assignments should reflect the demands of the game being made, and any assets produced for assessment purposes that are of production quality should be integrated into the game. Assessment Assignments should generally take one to one-and-a-half weeks to complete.

The Director is responsible for devising a way in which to quickly provide feedback to the team regarding performance on the Assessment Exercise. This often requires assistance from the sub-team Technical Directors, as team sizes are commonly too large for one person to provide feedback to in a timely manner. While the Director and Technical Directors are evaluating Assessment Assignments, all sub-teams should be working on learning the necessary technology to effectively complete their tasks over the course of the semester. Programmers should be researching UnrealScript and exploring the class hierarchies; Level Designers should be learning how to use the complex Unreal Editor program; Modelers should be learning the appropriate modeling and texturing techniques for production using Unreal 3; Animators should be exploring production standards for importing animations into Unreal Editor; and Concept Designers should be producing work at an accelerated rate for use by the other teams by the time they reach the end of their training.
In particular, it is recommended that Level Designers are evaluated at the end of their training period with a brief examination (see Appendix F), administered verbally to ensure that they are familiar with a baseline number of functions in *Unreal Editor* before they proceed with their design duties. If failed, the Level Designer should be given until the next class period to re-take the examination. The focus of the test should not be any sort of assigned grade, but rather a demonstration of familiarity and comfort with the software.

Also during the second week of class, a weekly meeting should be scheduled for the Director, Assistant Director, Art Director, Technical Directors and Research Leads to meet briefly and discuss the current progress of the project. Issues that arise in each sub-team should be addressed by their Technical Director and/or Research Lead; cross-team issues should be discussed, and asset requests should be made during these meetings. Generally these weekly meetings should not exceed twenty minutes in length and are tailored for releasing important information to the team quickly. During the first meeting, expected protocol for interaction between the teams and leads should be explored, and cross-team tasks should be discussed to ensure that all leads are aware of their responsibility to the team and to each other.

Following assessments, the Director should use the next few weeks of class to create a refined table of milestones and major due dates for all of the sub-teams and ensure that they are made public to the team. Seeing the end goal and how it relates to the smaller, incremental tasks leading up to it is important to maintaining unity of sub-teams as they work together over the course of production.
By the time training and other administrative tasks are completed, the Chico State Game Studios team should have a solid understanding of the tasks necessary to reach a final product. It is important that the project gain a certain rhythm and momentum, with consistent milestones and a steady stream of tangible asset production. This momentum allows the Director some time to set up other background tasks, such as outsourcing a promotional website for the game or establishing a working relationship with the largely external Writing team.

Deadlines during production should occur on a weekly basis to ensure that assets are being produced at the necessary accelerated pace throughout the project. For example, during the production of *High Noon* and *D.A.V.I.S.*, Modelers produced between two to four completed 3D assets per week while Concept Designers produced two finalized model sheets per week. Maintaining consistent deadlines throughout the project allowed students to balance their work for Chico State Game Studios with the demands of their other courses and was integral to preserving the quality of assets.

Throughout the semester, it is the duty of the Director to ensure that the project adheres to major established deadlines. If deadlines are missed, the Director should re-evaluate the schedule to project how the missed deadline will affect others in the future. It is also the Director’s responsibility to modify the feature list for the game if the schedule is jeopardized. For example, to complete the *High Noon Unreal 3* conversion in a timely manner, two weapons and three enemies were cut in the transition from the original *Unreal 2004* implementation. This allowed the team to focus on making the remaining weapons and enemies higher quality and bug-free.
The Director and Technical Directors are also responsible for carefully monitoring the sub-teams to verify that all team members are receiving an appropriate amount of work to complete during their time on the project. It is important to limit or eliminate downtime between assignments on any sub-team so that momentum is maintained and that time is spent actively creating assets. Conversely, if a team member is overwhelmed their workload should be immediately adjusted.

Quality assurance should be actively pursued across both semesters of production. Quality assurance includes verifying that all produced code and art assets reach a baseline level of quality before they are implemented in-game. In the current Chico State Game Studios organizational structure, there are a number of built-in checks to ensure that multiple team leads view a given asset before it can be considered production quality (see Figure 10). All Technical Directors serve as the first check for all

![Quality Assurance Checks for Game Assets](image)

*Figure 10. Quality assurance checks for game assets.*
of the sub-teams, providing iterative feedback during the creation of any given asset. Art related assets are usually then checked by the Art Director, and in some cases passed to the Director for final clearance. Game assets that do not fall under the Art Director’s jurisdiction are checked by the Director after they pass through the appropriate Technical Director. Technical Art is slightly different, being checked by the Technical Art Lead, the Assistant Director, the Art Director, and finally the Director for approval. These additional checks exist to account for the strong influence exerted by Technical Art on the final aesthetic implementation of the game.

At the mid-term and final weeks of the semester, the Director should administer Mid-Term and Final Self Evaluations (see Appendix I) to gauge how each team member feels they are performing in the class, including: attendance, work load, hours per week dedicated to the class, and rate at which work is being produced on schedule. Also surveyed is each team member’s satisfaction with the production pipelines in use, and an open invitation to communicate any ideas they may have to improve the class. These evaluations are vital in maintaining an open and safe means of communicating with each individual student about sensitive information such as personal performance and letter grades.

By the end of the first semester, the game should be in an Alpha state, meaning that all core game functionality has been implemented and may enter a testing phase. Finalized art assets may or may not be visible in-game at this point, but the game itself should be playable at a core level. Bugs may be present in the code at this point, and many of the implemented features might be rough or serve as placeholders for future implementations. Playtesting should begin during the end of the first semester of
production, checking new functions as they are implemented to ensure that they are still compatible with the game, and most importantly, still fun.

The first semester of production concludes with a public presentation, called the Semester Release, which follows the annual Excellence in Computer Graphics Awards. Organized and delivered by the Director and Assistant Director, this presentation, usually forty-five minutes in length, features the accomplishments of the Chico State Game Studios team for that semester, as well as a detailed plan for the following semester of production. The Semester Release should feature a demonstration of the in-progress game, highlighting the significant achievements of the team thus far.

**Summer**

The Director should use the summer months before the next semester of production to review the progress from the previous semester and itemize what tasks need to be completed to bring the project to Beta release status. Like the Preproduction Semester, a prototype schedule for the next semester of production should be assembled based on the rate of progress during the previous semester, including projections based on results of the completed Final Self-Evaluations.

Additionally, it is recommended that the Preproduction team continue to meet prior to the second semester of production to make any needed changes to the story, characters, or game discovered during playtesting. Also during this time, the wiki version of the game design document should be revisited, ensuring that all information presented therein is current and accurate. Any changes to the game should be reflected in the wiki and distributed to the Chico State Game Studios organization at the start of the new production cycle.
Also during summer, the Director may have volunteer staff that wishes to contribute to the game while classes are not in session. Because production is technically not active, it is the duty of the Director to ensure that all content produced during the summer months is consistent with the existing content and the future plans for the game.

If the game has special production requirements such as voice acting or video, it may be wise for the Director to contract outside help to complete these tasks while the rest of Chico State Game Studios is essentially on hiatus. Without the major in-house teams actively producing content, the Director is free to troubleshoot and brainstorm with outside teams to create content that can be used and integrated by the core Chico State Game Studios team when production resumes in the fall.

Continued Production and Post Production: Semester Two

The second semester of production on a Chico State Game Studios project begins in much the same way as the first. A syllabus must be created detailing the requirements of the class and the manner in which all team members will be graded, team policies should be explained in detail and the structure of the Chico State Game Studios organization should be covered to ensure that all team members understand the nature of the organization, particularly any structural changes that may have occurred from the first semester to the second.

Generally, the Chico State Game Studios roster changes significantly between semesters as students graduate or encounter course conflicts with the Chico State Game Studios class. Therefore, regardless of the number of return students, the first two weeks of any Chico State Game Studios production are nearly identical every semester. New
teams must be organized, new Technical Directors and Research Leads established, and new assessments and training exercises assigned to prepare the new team for production. For returning students who do not necessarily need to be assessed again, it is common that they will simply resume production where they left off at the end of the previous semester.

Production should not change greatly for most of the major teams: Modeling should continue to produce 3D assets at a rapid pace; Animation should continue refining old animations and creating new ones; and Programming should be refining game functionality and frequently testing for bugs. The external Writing team will likely experience a small decrease in workload, but will largely work as they did the previous semester: producing content for dialogue, in-game written materials such as newspapers and journal entries, and authoring the game manual. The Audio team should expect an increase in workload, as the integration of audio is widely a post-production activity.

It is recommended that returning Level Designers continue work on their previous semester’s levels; new Level Designers placed on old levels will require extra time on top of training to familiarize themselves with the work of the previous semester’s designer. If the Level Design team has a one-hundred percent return rate, then it is recommended that no new Level Designers are brought onto the team, unless the project requires it.

The only team that incurs a significant structural change is the Concept Design team, who should ideally be finished with the majority of their work relating to establishing the visual style, identity, and period research for the game. While model sheets do continue to be produced regularly to maintain a constant 3D asset flow through
the Modeling team, many Concept Designers move on to creating more formalized, cinematic images for the game. During *High Noon*, these images were used primarily for expanding the rich game mythology to increase team unification and for use on promotional materials such as websites, art books, and game packaging. While producing still images according to game mythology may seem like a secondary task, the positive effects of generating these fun, creative, professional pieces on the team’s understanding of the game were noteworthy. Concept Designers will likely also spend a great deal of their time producing in-game artwork such as billboards, advertisements, murals, and other items that require original, high quality, artwork according to the established visual style.

The Director spends the first few weeks of the second semester of production reviewing the previous semester’s work and verifying that they still adhere to the production standards established during the first semester. Lower-quality or non-production quality assets are sent back to the Modeling team for revisions, animations are frequently revisited, and code is checked for bugs and reliability.

All milestones for the second semester should be focused on getting the game to Beta status. A game that is in Beta is nearly entirely implemented in terms of functionality and art assets, and most severe, game crippling bugs have been located and resolved. With that in mind, the semester should be spent finalizing all code written for the game, polishing assets, and ensuring the game is playable and fun. Playtesting should occur regularly during the second semester of production both in and outside the classroom. Gamers unfamiliar with the project should be invited to participate in external, observed playtests to get accurate readings on how the end-user will interface with the
game. Changes to the game based on playtesting should be small and iterative, building slowly off of one another and tested to gauge how the changes affect the game overall. Additionally, the integration of any post-production elements such as video, cinematics, special effects, and special scripted camera events should occur in this semester.

Like the semester before it, the now completed game should be released at a public presentation during the last week of classes. If the game is in true Beta, it should be uploaded to the Chico State Game Studios website for distribution, opening the game to the public for more advanced playtesting. The presentation, again lasting for approximately forty-five minutes, should showcase not only the two semesters of work but the game itself in its playable form, explaining the processes used to make it, the improvements made on the game from the previous semester, and the plans for the game once it is formally Beta tested. All further developments on the game are voluntary, as Chico State Game Studios prepares to work on a new game the following semester.

The Director’s last responsibility is to author a post-mortem of their work on the project, detailing successes and failures of the game’s production, the pipelines used and how they were modified from the previous project, and recommendations for future Chico State Game Studios Directors. This post-mortem should be between ten and fifteen pages and be made available to all Chico State Game Studios students via the Chico State Game Studios website.

Chapter IV will cover a number of recommendations and anecdotes for working with a Chico State Game Studios pipeline, including the importance of certain aspects to the success of a large-scale video game production project. These
recommendations should be used as a supplement to the pipeline, team organizations, and procedures outlined in this chapter.

Chapter Summary

This chapter provides an in-depth exploration of all roles and relationships in the Chico State Game Studios organization on the individual and group levels. Additionally, this chapter provides a historical recount of each of the three semesters of *High Noon*’s development and *D.A.V.I.S.*’s preproduction semester. Lastly, this chapter provides the reader with a semester-by-semester breakdown of how Chico State Game Studios can be successfully run by implementing a Three-Semester System and introducing new projects in such a way that they overlap the old ones. This chapter is expected to be of significant value to incoming Chico State Game Studios directors as it highlights in particular their duties to a game production from concept to completion.
CHAPTER IV

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

This chapter is designed to outline in detail the many changes to the Chico State Game Studios organizational structure and production pipelines that occurred during the High Noon and D.A.V.I.S. projects. In addition to justification, the negative and positive effects of each change will be explored. Recommendations will be made based off of these analyses for future Chico State Game Studios Directors and other students managing large-scale video game production projects.

Changes Made to the Chico State Game Studios Team Structure

Chico State Game Studios required a number of significant structural changes over the three semesters of High Noon’s production because of significantly increased team-sizes and next-generation engine demands. New positions such as the Art Director and Research Lead were added to the team to account for quality control and better distribution of responsibility. Two sub-teams were added to Chico State Game Studios during High Noon, the Audio and Writing teams, to account for increasing project demands for sophisticated music and dialogue. Existing roles such as the Technical Director and Assistant Director were redefined, increasing their responsibility to the team and providing additional oversight to the sub teams. Lastly, experiments in combining
sub-teams of different disciplines were conducted, yielding both positive and negative results.

**Team Restructuring**

Prior to *High Noon*, Modelers working for Chico State Game Studios were not required to UV-map or texture their models. Often, Concept Designers were asked to texture most of the in-game assets. At the suggestion of former Chico State Game Studios directors and friends in industry, with the start of production of *High Noon*, Modelers became responsible for their models from start to finish: researching, modeling, UV-mapping, and texturing all models generated for the game.

It should be noted that in the first semester of *High Noon*’s production, Concept Designers were asked to contribute to texturing Second-Tier, or lower priority assets. While it may seem very logical to move artists into texturing when there is a shortage of team members who are proficient digital painters, this policy was never successful for *High Noon*, nor was it successful for previous Chico State Game Studios projects. Concept Designers were typically moved to texturing late in production, giving them little time to learn and understand the texturing production standards for the project. In addition, painting for conceptualization and painting for texturing are significantly different undertakings. Painting for conceptualization involves a great deal of traditional art techniques including perspective and shading, whereas painting for texturing is a significantly more abstract process. Because of this, most of the textures generated by the Concept Designers in the first semester of *High Noon* had to be redone in the following semester because they did not meet quality standards. Not only that, but because of the unfamiliar nature of the assignment, most Concept Designers were unhappy texturing,
finding it to be an uninspiring and difficult endeavor. Morale on the Concept Design team dropped significantly in the end of the first semester of *High Noon*, and veterans of the first semester currently working with Chico State Game Studios still shudder when texturing is suggested for the Concept Design team.

It became clear during *High Noon* that Concept Designers should be moved to texturing only if they felt comfortable doing so, and that relying on Concept Design as a failsafe backup for texturing shortages generally had a negative effect on the team and project as a whole.

Initially, it was expected that adding so much responsibility to the Modeling team would significantly slow the process of asset generation. While there was a considerable learning curve, by the time *High Noon* hit its second semester of production most Modelers possessed the necessary skill sets to produce assets accurately and quickly. In comparison with the confusion that passing on the assets generally caused, keeping the models with the same artist for a longer period in the production pipeline minimized design mistakes, and overall created higher-quality products that adhered better to the art direction for the game.

An unexpected byproduct of resolutely requiring the Modeling team to learn to UV-map and texture was the creation of a team of experts. As Modelers were required to live with their work through three major milestones in the production pipeline, many took extra care to learn best-practices to minimize revisions and established methods for modeling quickly and accurately. Many of these Modelers rose to more senior positions during subsequent semesters of *High Noon*. 
In addition to adding responsibility to the modeling team, strict production regulations were applied to models and textures. Models were formally broken up into “First Tier” and “Second Tier” assets. A “First Tier” asset was usually an asset that the player would see often in-game and likely be very close to, thus requiring a higher level of detail than those that decorated the environment. “First Tier” models included primarily characters, enemies, and weapons. “Second Tier” assets encompassed the rest of the assets generated for the game; generally of lower polygon counts and possessed lower-resolution textures. Each assigned asset had a maximum polygon threshold and texture resolution specific to the type of asset. These regulations were strictly enforced by me and the acting Technical Director of Modeling with great success. Besides putting strict quality standards on the models generated for High Noon, modeling checklists (see Appendix G) were generated, file naming conventions and directory structures established, and modeling best-practices were taught in-class.

In an experimental move, the Concept Design and Modeling teams were merged into the same department for the High Noon project, becoming the Art Assets team. While this change in structure was generally very successful as it unified the two primary art generation teams, there were a few noteworthy drawbacks to the merger that required further restructuring at the beginning of the D.A.V.I.S. project.

The two largest sub-teams in a Chico State Game Studios project are generally Concept Design and Modeling. Merging the two largest sub-teams created a single sub-team that accounted for around one-half of the total team. The Art Assets team was managed by one Technical Director who usually specialized in modeling and generally lacked the vocabulary to communicate with the Concept Designers. In addition, the Art
Assets Technical Director often found his time impacted, frequently spread thin across the many team members under his supervision. As a result, I found myself regularly interfacing directly with the Concept Designers: creating assignments, managing incoming artwork, critiquing outside of class, teaching new techniques, and ensuring that all concept art generated for the class was of an appropriate quality. By the end of *High Noon*’s production in fall 2008, I realized what I was doing was essentially acting as an Art Director; a position that previously did not exist in the Chico State Game Studios organization.

Because of this, at the start of the production of *D.A.V.I.S.*, the role of Art Director was introduced into the Chico State Game Studios team organization. The Art Director is formally a member of the Art Assets team, and works alongside the Technical Director of Art Assets. Whereas the Technical Director typically oversees the aspects of art generation that directly interfaces with the game engine, the Art Director checks art for style consistency and delivers instruction on painting and texturing in relation to the visual standards and themes in the game. The introduction of the Art Director has been particularly successful in alleviating pressure on the Technical Director and adding an additional level of quality assurance to the art generated for the project. It is expected that the addition of the Art Director to Chico State Game Studios will help alleviate the shortages in texturing personnel that have plagued projects in the past.

Also among the noteworthy additions to the Chico State Game Studios team format was the Research Lead position, which was implemented in the third semester of *High Noon* and created as a response to the decision to use the new *Unreal 3* engine. The Research Lead is designed to alleviate the Technical Director of some responsibility by
being in charge of finding best-practices and solutions to recurring technical issues on a
given sub-team, and most importantly by documenting those solutions for the sub-team.
The Research Lead is also responsible for the propagation of information through their
sub-team and ensuring that problems are solved quickly. A Research Lead is assigned to
each major Chico State Game Studios sub-team (Art Assets, Animation, Level Design,
and Programming) and expected to become a specialist on their given team. The
Research Lead position proved to be a suitable compliment to the Technical Director
position, and began to mitigate the previously existing issue of lack of documentation on
Chico State Game Studios projects.

In addition to the Research Lead and the Art Director positions, the changes in
the production pipeline brought about by using a next generation engine during the third
semester of High Noon presented the need for a new sub-team: the Technical Art team.
The Unreal 3 engine includes a sophisticated materials editor that supports many next-
generation surfacing features, allowing us to create realistic metals, fires, rusts, and other
environmental effects at higher resolutions and with better algorithms than with the
Unreal engine’s 2004 counterpart. However, creating the materials in Unreal 3’s editor
took not only significantly more time than in Unreal 2004’s editor, but it also took an
individual who was highly trained both in the Unreal 3 Editor and in the generation of
various types of specialized texture maps. The Technical Art team was tasked with taking
models and textures generated by the Modelers, importing the models into the game, and
setting up complex material nodes to achieve the appropriate type of surface on the model
(normal maps, specular maps, reflection maps, et cetera), and testing them in-game. In-
game assets were then bundled by the Technical Art team into packages and sent to the
Level Design team for placement in levels. The addition of the Technical Art team allowed *High Noon* to achieve a professional and relevant level of sophistication in the visuals presented in-game, making *High Noon* look truly “next-gen.”

**The Role of the Assistant Director**

One of *High Noon’s* greatest structural failings was the lack of a formal Assistant Director. During the first two semesters of *High Noon*, the Assistant Director essentially acted as the prototype for the Research Lead position, and was charged with exploring implementation methods for complicated gameplay elements such as physics and particle emitters. During *High Noon’s* third semester of implementation, the Assistant Director role was never filled, leaving what responsibilities the Assistant Director did have in the hands of the Director. It is for this reason, in the *High Noon*-based Chico State Game Studios team structure, that the Assistant Director essentially has no role.

*High Noon*, by its third semester, grew to fifty-two people working actively on the project. With no Assistant Director, it was the responsibility of the Director to not only oversee the day-to-day operations of each team, but to manage all assets being produced by each team while maintaining administrative elements such as the design document and individual and team assignments. With a team as large as *High Noon* was, this became an overwhelming amount of work and tasks frequently got lost in the whirlwind of rapid-paced production.

In particular, defaulting to the Director to oversee asset management created a large number of problems in regards to version control and asset tracking. Because of the large number of assets coming through the pipeline at any point during the project, it
became increasingly difficult to ensure that those assets were quickly and accurately passed to the appropriate sub-teams. Duplications, lost assets, and frequent mistakes occurred as a result, costing the project significant time. In addition, beyond the Director, there was no one explicitly responsible for asset quality control, and frequently assets made it to the end of the modeling pipeline that were significantly below quality standards for the project. It became clear that the team would need a dedicated member to handle the organization and categorization of assets, as well as serve as a significant quality control checkpoint.

The Assistant Director is well-suited to provide all of these things. Usually a senior member of the team, the Assistant Director should be familiar with the technology used in the class and have a strong understanding of the production pipeline. In addition, the Assistant Director works closely with the Director on day-to-day production operations and clearly understands the Director’s policies and strategies for the project. Whereas the Director can serve in a global capacity to the team, the Assistant Director can serve in a local capacity, overseeing day-to-day administrative/clerical tasks and providing organization and structure to the content generated by the sub-teams.

Though *High Noon* never properly implemented the role of the Assistant Director, the role was formally established in the Chico State Game Studios team organization by the beginning of the implementation of *D.A.V.I.S*. The responsibilities of the Assistant Director should include:

- Serving as a proxy for the Director when the Director cannot be present.
- Managing assets by carefully tracking and categorizing all assets generated for the project.
• Maintaining the pipeline by ensuring assets are moving swiftly and to the correct sub-teams.

• Providing a quality assurance checkpoint by carefully reviewing assets to ensure they adhere to quality standards for the project.

• Clerical tasks relevant to the project.

The implementation of this support role is expected to greatly increase productivity on the Chico State Game Studios team by minimizing quality errors and reducing redundant work by ensuring assets are closely tracked.

The Role of the Technical Director

The role of the Technical Director was particularly difficult to define over the course of the *High Noon* project. Originally, a Technical Director for Chico State Game Studios was appointed as a sub-team lead and had a number of micro-responsibilities directly to his or her team. According to Chico State Game Studios precedent, all sub-teams are appointed one Technical Director. However, the responsibilities of the Technical Director seemed to change from project-to-project. When production for *High Noon* started, I found myself tasked with defining the role in terms of *High Noon* specifically.

This task was initially difficult because we had no clearly developed pipeline for a Chico State Game Studios project at the start of *High Noon*, so the typical tasks appropriate for the Technical Director to manage were not previously defined. In addition, very few team members were qualified or willing to serve as a Technical Director during the first two semesters of *High Noon*. As a result, the responsibilities of a Technical Director were initially quite minimal, consisting of providing brief weekly
updates on the state of the sub-team and helping team members with technical issues in-class.

Because the Technical Directors were not used effectively during the first two semesters of *High Noon*, direct management of all teams fell on the shoulders of the Director. I directly corresponded with each team about assignments, due dates, expectations, troubleshooting, technical issues, et cetera, as well as provided critique for assignments and in-class lessons on best practices. This level of micromanagement was naturally impossible to maintain as the project progressed, and consequently many teams did not receive an appropriate level of support to keep quality and productivity at acceptable levels, costing time and resources.

The negative effects of the undefined role of the Technical Director continued to escalate as the *High Noon* team size grew from its modest eighteen-person beginnings. In the third semester of *High Noon*, the addition of formal team education to the curriculum and the subsequent increase in skill sets allowed us to restructure the role of the Technical Director. Technical Directors became considerably more involved in the management of their teams than in previous semesters, alleviating most of the need for micromanagement of any kind on the part of the Director. By the third semester of *High Noon*, the duties of the Technical Director included (which in turn extended to the Art Director in *D.A.V.I.S.*):

- Checking in with individual sub-team members and providing the Director with a brief progress report weekly at minimum.

- Ensuring all sub-team members have a current task and are making efforts to complete it.
• Ensuring all sub-team members have the resources they need to complete the tasks assigned to them.

• Supporting the Research Lead in the dissemination of knowledge through the sub-team.

• Attending a weekly meeting with all sub-team Technical Directors, the Director, and the Assistant Director to recap the week and cover any major outstanding issues.

• Administer tutorials on team best-practices for relevant software and production techniques.

• Maintaining out-of-class communications and critique with the sub-team via e-mail and instant messaging software.

Restructuring the responsibilities of the Technical Director was highly beneficial for the *High Noon* team and significantly increased productivity. Sub-teams began to work as tighter-knit units, maintaining closer communications than in previous semesters and acting more effectively as a team.

**Pairing Sub-Teams**

*High Noon* formally brought about the pairing of sub-teams to accomplish complicated tasks. During the first two semesters of *High Noon*, Concept Designers and Modelers were frequently paired for rapid asset generation. This allowed Modelers to work closely with Concept Designers to request specific designs for complicated assets, as well as make quick changes to designs if necessary.

Pairing one Concept Designer to one Modeler was an interesting idea in theory, but in application it was not without its complications. Often, miscommunications
happened between the Modeler and Concept Designer that resulted in incorrect designs. Because the team lacked an Art Director (which was introduced in the preproduction phase of *D.A.V.I.S.*), the designs were not actively being checked for quality against the rest of the art generated for the game, and would often make it to final model before errors in the design were caught. Overall, the policy of pairing Concept Designers and Modelers failed for *High Noon* and cost the project valuable time and resources.

Lost time was primarily attributed to the fact that the Concept Design team was not formally overseen by anyone but the Director during the production of *High Noon*. On paper, it was the duty of the Technical Director of Art Assets to oversee both Modeling and Concept Design teams, but he rarely had time to manage all of the Modelers, much less anyone else. As Director, I also found myself with very little spare time for directly managing the Concept Design team. As a result, the Concept Design team essentially became an outlier team, reporting directly to me on occasion but essentially working independent of the rest of the team.

In another team-pairing experiment, Programmers were matched with Level Designers during the *Unreal 3* implementation of *High Noon* in its third semester, as it was rare that Level Designers had any sort of prior programming experience and often level-specific scripts had to be generated for quest functionality. The Level Designer/Programmer relationship was symbiotic; Level Designers designed levels from start to finish and worked collaboratively with Programmers in implementing all dynamic functionality in-game.

During the first two semesters of *High Noon*, the implementation of quests was abstracted from the Level Design team and delegated to the Programming team. This
 level of abstraction greatly slowed down the quest implementation process and seemed to create a disconnect between the environmental design of the level and the gameplay design of the level. Overall, level objectives implemented during the first two semesters of *High Noon* were incongruous with the environments in which they existed. Most importantly, pushing quest implementation via Programmer to the end of the pipeline proved to be a critical mistake because it allotted no time for playtesting.

The success from combining Level Designers and Programmers was instantaneous. Levels were developed quickly and accurately with the quests being implemented side-by-side with the levels themselves. As a result, levels were developed iteratively, with nightly builds being available to the team for playing and exploring. Being able to track nightly progress allowed us to quickly identify flaws in quests, as well as identify quests that were simply not fun once they had made the transition from paper to electronic game. Pairing Level Designers and Programmers allowed us to create more complicated, intricate game sequences than we had been able to make in previous semesters, bringing a new level of sophistication to *High Noon*. Moreover, mandatory weekly playtesting sessions were implemented, which significantly increased the overall quality and playability of our levels.

Requiring Level Designers and Programmers to work closely together throughout the development process also emulates more closely the methods and practices of professionals in the game industry. In addition, the importance of reaching playtesting early in development is heavily stressed in industry, and yielded undeniably positive results when it was implemented during the third semester of *High Noon*. 
Changes Made to Chico State Game Studios Policies

Major policy changes occurred both prior and during the *High Noon* project in regards to how the class was administered academically and to the priorities of the team as a whole. Instead of completing projects over the course of one academic year (two semesters), an additional semester was added to allow sufficient time for preproduction. Instead of being structured as a Credit/No Credit special topics course, Chico State Game Studios became a 3.0 unit graded course with required attendance. Communication, team education, and interdisciplinary collaboration became the foremost goal of Chico State Game Studios, and the act of learning to create games took priority over the completed game itself. Overall, the changes made to Chico State Game Studios policies added much-needed structure and recognition of the organization’s academic roots.

Introduction of the Three-Semester System

The introduction of the Three-Semester System into the standard practices for Chico State Game Studios was a response to *High Noon*’s frequent issues with late additions and changes to the design of the game. The Three-Semester System (Figure 5.1) was designed to mitigate last-minute decisions and ensure the design of the game stays consistent with the Director’s original vision by adding a semester dedicated solely to preproduction.

Prior to the introduction of the Three-Semester System, Chico State Game Studios allotted two semesters per game production. Preproduction was not addressed in the two-semester system; instead, directors were required to design their game over the summer preceding the first semester of active production. Directors had the option of
designing alone or with the assistance of willing student volunteers. Designing over the summer was usually met with varying levels of success, primarily because of the difficulties involved with organizing students during a time when school is not in session.

Under the Three-Semester System, two game projects run concurrently during the fall semesters, with a new game starting preproduction every fall semester. New projects, which are pitched by students in late April, are selected and announced every spring semester. New directors have the option of using the summer months between terms on additional preproduction, but are offered time and course credit for preproduction in the following fall.

This system ensures that the studio utilizes the entire team at all times during production, and that there are no slow periods between productions. Using this model, Chico State Game Studios is effectively in full-time production on a video game project every semester.

For example, it is common in the third semester of a project’s development that Concept Designers are no longer actively needed to generate art for the project. Under the two-semester system, Chico State Game Studios struggled to find work for the art team while production was coming to a close. Under the Three-Semester System, Concept Designers can be moved gradually to the incoming project as they conclude their work on the active project. This was met with great success during the preproduction semester of *D.A.V.I.S.*, providing the Chico State Game Studios team with a smooth and natural transition into the new project the following semester.
Figure 11. The three-semester system, illustrated.

The tasks that should be completed during the preproduction semester include, where applicable:

- Establishing the story.
- Developing and testing important gameplay elements.
- Development of a comprehensive game design document.
- All written in-game materials generated, including: quests, books, maps, newspapers, character dialog, et cetera.
- All voice sound-bytes recorded.
- All video filmed for in-game footage.
- Development of puzzles and puzzle solutions.
- Contracting music.
- Rough level blueprints.
- Environmental concept art.
- Character concept art.
- Paper prototyping and introductory playtesting.
Because completing these tasks quite often requires a herculean amount of dedication and organization, the formation of a small creative team is highly advised during the preproduction semester for the incoming game project. This team should ideally be composed of around four senior-level students who have a strong understanding of the game development process, and the incoming Director and Assistant Director. It is advised but not required that the active Director serve on this committee. It is also advised that specialists in programming and art be present on the committee to ensure that ideas generated are feasible in the chosen game engine. Designing by committee, while generally a slower process than designing with a single designer, lends itself well to ensuring that the gameplay, plot, characters, puzzles, and other crucial game elements are well-designed and consistent with the rest of the design before the game goes into production.

In addition to catching defective gameplay devices and plot holes, the preproduction phase of the Three-Semester System is also very important to the new student Director. Under the previous two-semester system, the new Director was not formally trained before taking over the incoming project. With the preceding project still in production, the new Director has instant access to the acting Director’s expertise and advice, and can learn alongside the acting Director during their own preproduction work.

The incoming Director not only has the acting Director as an on-call resource, they also have a semester to adjust to the demanding role of directing. This is beneficial not only for the Director but the incoming team, both of whom can ease into the new personalities and management styles.
Despite its many successes, there are some noteworthy potential drawbacks to the Three-Semester System. Foremost of these drawbacks is the amount of strain it puts on the Chico State Game Studios team in terms of personnel; as the new project is introduced into the organization, resources become spread over two projects instead of one. Additionally, the small creative team working on the new project is likely to be composed of senior-level students, many of whom have significant experience working in game development and are usually Chico State Game Studios veterans. Moving these people from the active project to the incoming project can leave the active project vulnerable if the remaining team members cannot offer the skills and expertise of the missing team members. This risk is ideally mitigated by the Research Lead position on each of the major teams.

The Three-Semester System also does little to formally protect the delicate nature of the preproduction phase, and requires a great deal of structure and work outside of the classroom on the part of the small creative committee, as the focus of in-class time for Chico State Game Studios during this semester is the active project and not the incoming one. It is important that the incoming Director have a strong understanding of scheduling, making and meeting deadlines, and organizing frequent creative meetings during the preproduction development process. Failure to adhere to a strict schedule of milestones could result in the failure of the project early on, with mistakes and oversights made during preproduction propagating through the following two semesters of production.

The preproduction semester was formally tested with favorable results for Chico State Game Studios’ fifth game production, *D.A.V.I.S.* During the preproduction of
D.A.V.I.S., a small creative committee was formed, consisting of the current Director and Assistant Director, an experienced artist, myself (as a former Director and Art Director), and a senior Programmer. This team proved to be ideal for catching potential pitfalls and problems early in development, as well as providing interesting creative angles on plot and character development. Additionally, the diverse technical and creative backgrounds provided by the committee members brought many new perspectives and ideas to the game’s design, which greatly contributed to a more sophisticated, multi-faceted, and fuller design by the end of the design phase.

Creative meetings were held weekly for fifteen weeks, lasting between two and six hours per meeting. Committee members frequently completed small homework assignments outside of class that kept the creative process from stalling during meetings. The game was developed from basic concept to complete design document during this time, ending in a culminating presentation to the public about the game and production plans for the following semester.

**Required Attendance and Grading**

With the introduction of *High Noon* came a fundamental change in the manner in which Chico State Game Studios was conducted: changing the class from a Credit/No Credit class to a graded, 3.0 unit course. While working on a Chico State Game Studios project, students are now enrolled in APCG 495 *Advanced Production*, which has formalized lab and lecture times.

Attendance to Chico State Game Studios lab and lecture times became mandatory in *High Noon*, and was enforced daily by calling roll and giving credit for being present. Class policy specified that classes could be missed if cleared with the
Director beforehand, but that regular attendance was crucial for the level of communication and cooperation that the team would need to be successful on the project.

Making the class graded and requiring attendance adds a level of structure to the course that is necessary for the large caliber projects on which Chico State Game Studios typically works. Students understand the non-negotiable time commitment to the class from day one, and the act of communicating standards early on typically mitigates violation of those standards as the project progresses.

**Implementation of the Mid-Term and Final Self-Evaluation**

*High Noon* introduced the active use of the Mid-Term and Final Self-Evaluation (see Appendix I): one page simple assignments that required students to assess their own individual performance in the class in accordance with questions regarding the number of hours spent outside of class on the project, number of absences, ability to keep up on assigned workloads, et cetera. Students were aware via the course syllabus (see Appendix D) that the completion of the Mid-Term and Final Self-Evaluations accounted for 50% of their calculated grades in the course, with the other 50% being an equal distribution among attendance and Director Evaluation. Self-Evaluations were completed for all three semesters of *High Noon’s* production.

The Self-Evaluations became a significant means of not only statistically tracking the students’ perception of the progress of the class, they also served as a means for students to communicate their ideas about how the class could be better run, or to share what they thought was successful about the class or project.
Self-Evaluations appeared to empower the students to provide feedback that they knew was going to be carefully read and considered in the classroom environment. Comments were generally helpful and thoughtful, and overall students offered useful critique that helped the future implementations of the class. The integration of the Self-Evaluation into the Chico State Game Studios curriculum added to the openly communicative culture that *High Noon* fostered and proved to be an excellent contributor in maturing the class into a functioning student game studio.

**Mandatory and Encouraged Methods of Communication**

Communications both inside and outside the classroom were heavily stressed during production on *High Noon*. Rules for mandatory external communication between sub-teams on *High Noon* were enforced for all three semesters of the project’s life.

Utilizing many different methods of communication was heavily stressed in *High Noon*, with perhaps the most significant of them being the class Google Group. A Google Group is a free web application created by Google Inc. specifically for large group projects that require inter-team communication and file sharing. With the Google Group, students had instant access to important class documents like the syllabus and design document, as well as concept art, tutorials, and important class announcements. The Google Group doubled as the class mailer, allowing the administrative staff (Director and Technical Directors) to e-mail large groups with ease.

In addition to being networked through the Google Group, *High Noon* team members provided their relevant e-mail addresses, phone numbers, and instant messaging information for the class Team Contact Information document, which was also housed on
the Google Group’s files archive. This document, in combination with the accessibility
provided by the Google Group, made communication with most team members
practically effortless.

Instant messaging (IM) technology was used frequently for instantaneous
feedback to students and troubleshooting. Most popular IM protocols were used during
*High Noon*, including *GoogleTalk*, *AOL Instant Messenger*, *ICQ*, *MSN Messenger*, and
*Yahoo Instant Messenger*. The use of instant messaging technology allowed team
members to work collaboratively via the internet at non-standard hours from any
location. Overall, the use of instant messaging technology was a large contributor to the
successful level of communication between students and sub-teams in *High Noon*. For
large creative projects like *High Noon* where many decisions must be made
collaboratively, heavy reliance on instant messaging technology is greatly encouraged.

Students were also required to check their e-mail at minimum once per day as
it was the official means of communication for the course. As such, e-mail was used
heavily during the three semesters *High Noon* was in production. Unsurprisingly,
communications between me and my team alone attributed for over 2.5 gigabytes of
information sent via e-mail.

**Communication Through Visual Feedback and Presentations**

As previously stated, *High Noon* was composed of five major in-house sub-
teams: Concept Design, Modeling and Texturing, Animation, Level Design, and
Programming. To maintain a functioning production environment, then, communication
must not only happen on the micro-level with the sub-teams but on the macro-level with the team as a whole.

*High Noon* used the lecture component of the class as a means of presenting *weeklies* to the team. As the name implies, a weekly is a meeting that occurs once per week during production. At a weekly, sub-teams and individuals present their in-progress work and significant research to the Chico State Game Studios team and Executive Producer. The weekly used at Chico State Game Studios is inspired by the *daily* typically used in film production. Since Chico State Game Studios only meets three times weekly, a daily format was not feasible and thus the weekly was adopted.

Quite naturally, maintaining good communication between sub-teams of specialists can be a difficult task as it is unlikely that a Programmer, for example, will take valuable production time to walk down the hallway and catch up with the Concept Design team. Time spent working at Chico State Game Studios was therefore split into major foci: time with one’s sub-team and time with the team as a whole. The activity component of the class became dedicated to sub-teams, allowing specialists to work with one another in a production-centric environment. The lecture component provided an important venue for our specialists to learn about and understand the teams working parallel to them.

Weeklies were also useful in providing critique to the various art-based teams for in-progress assets. Weeklies were run as an open forum, encouraging feedback from all sub-teams in a free-form dialogue format. Critiquing in-progress work not only served as a successful means of quality assurance on the project, but it also educated the team about the processes used by the various sub-teams.
Likewise, when any significant milestone was reached, it was shared during a weekly. Creating a dedicated venue that allowed team members to see their work come to life in-game was invaluable for keeping up motivation, inspiring creativity, and generating enthusiasm for the project. Significant implementations were often met with abundant cheers and applause, and many team members who contributed to major game elements (music, weapon implementation, character designs, et cetera) were treated as minor celebrities in-class.

In addition to the visual feedback provided by the implementation of the weekly, *High Noon* was the first Chico State Game Studios project that published an art book. Designed to serve as a promotional piece for the game, *The Art of High Noon* was an 83-page, full-color celebration of the first semester of concept art and in-game models generated for *High Noon*. Published using Lulu.com, *The Art of High Noon* was available to the public for purchase at cost and was a great generator of enthusiasm among the Concept Design team.

**Stressing Team Education**

Chico State Game Studios prides itself on how closely it emulates the methods and practices of game studios in industry. Through a project, students are exposed to all major parts of a large-scale production pipeline, as well as encouraged to explore different jobs and responsibilities. Though Chico State Game Studios is essentially a class on large-scale video game productions, it has always been approached as a studio environment with the foremost priority being the completion of the project.

With *High Noon* came a shift in emphasis for Chico State Game Studios. During the first semester of *High Noon*, it was noted that most incoming students on the
project had never worked on a large-scale video game project before. As such, many students immediately lacked the skills and knowledge that they needed to be effective and productive team members. A significant part of *High Noon’s* first semester was spent educating the team on using the *Unreal* engine, as well as on proper modeling, animating, texturing, and programming techniques for the project. Because of this, full-fledged production on *High Noon* did not start until late in the first semester, and much of the work that was completed during the early phases when the team was still being educated had to be redone to meet established quality standards. It became apparent during this first semester and in subsequent semesters of *High Noon* that education would need to come first before the team could effectively emulate an industry studio.

I found this to be the foremost difference between Chico State Game Studios and game studios in industry during my work on *High Noon*. As much as Chico State Game Studios operates like an industry studio, it is important to remember that it is still housed in academia and operated by students. Few students are familiar enough with the technology used to make and design video games to create assets quickly and accurately. In fact, most students that participate in a Chico State Game Studios project do so to either hone a particular skill or experiment with a different part of the pipeline in hopes of finding their niche. That being said, rarely will you find a student who can immediately produce professional quality product on any of the sub-teams, because most students are still learning about the technology that they’re using, or looking to expand their skill sets by experimenting with a different team. It is important to note, then, that Chico State Game Studios cannot operate like a professional game studio until its “employees” can participate as professional team members. Unlike studios in industry, Chico State Game
Studios must first educate team members on the skills and theories that they need to be productive game makers.

In each subsequent semester of *High Noon*, approximately the first month of production became dedicated to assessments, tutorials, and education on procedure, pipelines, and other pertinent information to the project. Assessments are particularly of note, as they served as a preliminary assignment that communicated the level of comfort each student had with their team responsibilities. Based on the results of the assessment, subsequent assignments could be tailored to each team member’s skill set. This also greatly helped me plan the rate at which assets could be produced, as well as helped me identify lessons and lectures that needed to be delivered to the class on relevant techniques and procedures.

The time invested in team education on *High Noon* ultimately created a team of experts who in turn trained incoming students in following semesters. This had a positive effect on the project, with each team producing higher quality, more sophisticated work with each passing semester. In addition, each team worked faster than its predecessor, allowing us to create more content in shorter periods of time.

**Stressing the Interdisciplinary Nature of Video Game Design and Implementation**

Video game design, by nature, is an interdisciplinary activity. Games are composed of many sophisticated elements including music, sound effects, computer-generated visuals, animations, programming scripts, and integrated video. As such, specialists from many diverse disciplines are needed to create a video game from concept to final product.
During the first two semesters of High Noon, experts from other academic departments such as English and Music were not expressly utilized during production largely because of a lack of understanding of the production pipeline as it related to non-computer graphics elements. In short, the High Noon pipeline essentially lacked a realistic projection for elements such as voice acting, writing, sound effects, and music. As a result, the Unreal 2004 implementation of High Noon created during the first two semesters of production was essentially silent, and lacked any real implemented character development, and was generally absent of the rich back story and detailed characters that existed in the High Noon world.

In an attempt to improve player interaction with the game, by the third semester of production the High Noon team expanded to include experts in creative writing and audio: two writers to author in-game dialogue and cinematic sequences, two composers for music, and one sound effects artist for custom weapon, enemy, and environmental sound effects. The positive effects from this addition were instantaneous; music generation led to a greater understanding of the game’s tone and direction, writing breathed new life into the characters and reinforced the game’s whimsical attitude, and sound effects brought an undeniably gratifying quality to our weapons for the players. Adding these multimedia elements to the game unified the team as well, generating new ideas and renewed excitement for the project.

Game elements outside of the core functionality of the game such as music, video, and sound effects are easy to treat as an afterthought in game production, much as it was during the first two semesters of production on High Noon. In reality, games are a multimedia form of communication, utilizing many senses simultaneously to contribute
to the player’s interaction with the game world. Therefore, elements such as music and sound effects contribute to the player’s experience as much as the visuals, and should never be treated as afterthoughts. For Chico State Game Studios projects in particular, it is recommended that writers and audio experts are recruited early and included closely in the development of the game.

New Requirements of Chico State Game Studios Projects

Based off of research on companies in the industry and analysis of previously existing issues at Chico State Game Studios with regards to dissemination of knowledge and experience, two new practices are heavily recommended for every Chico State Game Studios project: the creation of a Post-Mortem report at the end of a project’s production, and the inclusion of frequent, targeted, iterative playtesting through the project’s life. These practices are designed to ensure that future Chico State Game Studios projects remain high-quality and professional, and that future teams are not plagued by the same issues as previous teams.

The Post-Mortem

The post-mortem is a report released at the end of a game’s production that analyzes the successes and failures of the team and production processes used for the project. Post-mortems are considered to be a standard means of analyzing game production processes in the game industry and are published frequently by large and small studios alike.

Lack of documentation has been a significant issue for Chico State Game Studios since its inception in 2004. No projects were formally documented beyond the
Design Document, and most Design Documents were many months out of date by the end of the project. *High Noon* even suffered from its own documentation problems due to times of frantic production and heavy crunch periods when documentation became an inconvenience in the face of completing the project.

Though Chico State Game Studios has worked on three titles prior to *High Noon* (*Chicago Crux*, *Bronze Age*, and *Viathan*), practically no documentation exists for any of the project’s production processes, policies, successes, failures, et cetera. Not only is there a lack of documentation, but it is rare that a student stays with the organization long enough to train newcomers. Chico State Game Studios by nature has a high personnel turn-over rate. As a Chico State Game Studios project is often considered to be a culminating activity for many students, most team members are seniors in the final years of their undergraduate educations and commonly graduate soon after joining the organization. The loss of valuable knowledge, therefore, is usually very high from project-to-project, with most projects starting with no existing knowledge-base. With no standardized methods of passing information on from project-to-project, new projects have essentially been stuck in a sort of nascence, unable to grow from the experiences and work of the projects that preceded them. I strongly believe this lack of a common knowledge-base has hindered all Chico State Game Studios projects thus far, and will continue to do so until the organization begins to require documentation on the part of the team and Director.

It is therefore strongly recommended that Chico State Game Studios adopt a policy of requiring at minimum a detailed post-mortem from each Director at the end of a project. Additional documentation from Technical Directors, Research Leads, and Art
Directors is recommended if possible. In addition and again at minimum, the Director post-mortem should be presented in a public venue as a culminating activity for the project.

It is expected that by requiring documentation and public dissemination of knowledge fewer mistakes will be repeated throughout the projects, alleviating the difficult transition into new projects and hopefully lowering the already steep learning curve.

Playtesting

“Playtest early and often,” stated Valve’s Kim Swift and Eric Wolpaw at their lecture, Integrating Narrative and Design: A Portal Post-Mortem presented at the Game Developer’s Conference in 2008, “Find out what your players actually want” (Swift & Wolpaw, 2008). Portal’s success, Swift and Wolpaw claim, is widely because they pushed playtesting early in the development cycle, and continued to playtest frequently as development progressed. This allowed them to constantly reaffirm that their game was still fun, and to carefully monitor the impact new additions had on the game.

Additionally, early iterative playtesting granted the Portal team many opportunities to monitor their game being played, allowing them to carefully study how players interact with their designs and adjust the delivery of content to better suit the player’s experience.

During the first two semesters of High Noon, playtesting was pushed until late in the development process because of massive bottlenecks in the level design and modeling production pipelines. Attempts at playtesting were generally disorganized and lacked direction and purpose, largely because of the unfinished and non-functional quality of most of the High Noon levels. Playtest sessions were therefore fundamentally
wasted, which in turn stunted the growth of our levels. Essentially, *High Noon* ran out of time during the end of its second semester, before it was ready to enter an iterative playtesting phase. This was widely due to a mismanaged asset pipeline and poor level design protocols during the first two semesters of the project.

Additionally, due to a massive oversight on my part, the *High Noon* production pipeline was not originally designed to include playtesting. Because of this, *High Noon* levels were not slated to be complete until late in the semester, and were not structured in such a way to allow for iterative testing. According to the *High Noon* pipeline, levels were playable when they were complete, which occurred close to when the game was considered complete. The repercussions of this serious planning error rippled throughout every phase of the project and naturally contributed to massive scheduling setbacks. The team was essentially never sure if their levels worked because they never allowed time to test them. Late in the second semester of *High Noon*’s production, an attempt was made to retrofit the pipeline to include mandatory playtesting sessions, but sadly it was so late in production that it had little effect on the functionality of the game. Because of this, while *High Noon* was a very beautiful game composed of four well-constructed levels at the end of the second semester of production, it was minimally playable due to incomplete scripting and severe game bugs: items that would have easily been identified and fixed early in production had a tight, iterative playtesting schedule been enforced during the project.

Using the recommendations presented in Swift and Wolpaw’s postmortem of *Portal*, the *High Noon* team pushed hard to get *High Noon* into playtesting early in its third semester. Changes to the modeling and level design pipelines greatly increased
flexibility in the assembly of the game and allowed us to pre-visualize important game elements before they were formally completed. As a result, we were able to playtest our levels before any formalized art assets were completed, using placeholder assets instead (see Chapter IV, *New Chico State Game Studios Design Strategies, The “Companion Cube” Approach to Level Design*) and allowing for Programmers and Level Designers to implement their level objectives far earlier in the pipeline than in previous semesters. Though *High Noon* still went over schedule in its third semester, its levels were exponentially more playable, visually entertaining, and fun; qualities that its previous-generation counterpart lacked.

Frequent iterative playtesting is crucial to ensuring the playability and fun of a video game. Video games are pieces of interactive software, and until the team knows how the user interfaces with the software, it cannot be known if the design is successful. Testing frequently and iteratively allows designers to make subtle changes to the game and to observe the effects those changes have on the player experience. To complete a game without testing it on the public to whom it will be released is to essentially release an incomplete game. It is therefore highly recommended that playtesting begin directly after the game is minimally playable and that playtesting continues frequently until the game has reached completion.

*New Chico State Game Studios Design Strategies*

Designing any game is a detailed, complex, and difficult process that often takes many iterations of experimenting, testing, and redesigning. Though many new strategies were experimented with during the course of *High Noon*’s production, the two
most effective design strategies used were providing constant emphasis on story and designing levels via “Companion Cube.”

**Emphasis on Story**

The decision to emphasize the importance of story-driven gameplay was an early one in the development of *High Noon*. Because of the scale of the project, I had concerns about keeping the art direction of the game unified. Additionally, I felt that we needed a mechanism to ensure that any new ideas and additions to the game did not compromise the fidelity of the core designs and plot of the original idea. Conversely, I did not want to present a game to Chico State Game Studios that did not allow for creative additions from the team. I strongly felt that the development of *High Noon* and the mythology surrounding the *High Noon* world should be a collaborative effort rather than the sole responsibility of the Director. Because of this, *High Noon* was designed specifically to give Chico State Game Studios team members wide creative control over additions to *High Noon* within a set of clear boundaries. New ideas were always encouraged as long as they did not conflict with the existing game world. I often referred to this as a *corralled* approach to design. The preliminary design document presented to the *High Noon* team contained a wide metaphorical fence around the borders of the *High Noon* world, and all team members were encouraged to propose additions to the plot, characters, and even quests as long as they fit within those borders and did not contradict any of the existing mythology.

Because of this approach, one of *High Noon*’s greatest strengths became its colorful and unique story. For the first time at Chico State Game Studios, the importance of story, context, and visual identity were constantly stressed throughout production in an
attempt to unify the team’s vision of the game and maintain the direction of the project. *High Noon*’s whimsical nature lent well to a healthy creative environment, encouraging strange gadgets, ridiculous puns, and inside jokes in the design of the game world. Designing for *High Noon* was unquestionably fun; rarely were we concerned with real-world functionality, but rather that game elements looked “cool” and were Steampunk enough for the game. The momentum created from this lighthearted approach to design carried strongly through the entire life of the project.

As a result, *High Noon* undeniably had the strongest identity and presence of any Chico State Game Studios game to date. Over time, the world in which *High Noon* took place and the characters that lived there took on their own identities; the villain of *High Noon*, Boz Novak, developed such a strong fan-base that our Concept Designers started developing fan art and custom back story for him, independent of the project. Suddenly the team knew Novak as a person, which was a particularly momentous achievement because of the bland treatment Novak was given in the preliminary design document, which basically described him as “the evil bad-guy.”

**The “Companion Cube” Approach to Level Design**

During the first two semesters of *High Noon*, the level design pipeline consistently hit serious bottlenecks while waiting for models and other assets to reach a level of completion appropriate for integrating into the in-game levels. This caused a number of problems during the development of *High Noon*; though Level Designers were encouraged to spend down time between asset packages fixing lighting and testing the level for appropriate flow and logic, many Level Designers found it difficult to envision
their levels without art assets, and production consequently stalled. *High Noon* levels remained practically empty until late in the second semester of production. This in turn affected playtesting, as it was practically impossible to playtest empty levels.

Recognizing this as a serious problem, the level design pipeline was redesigned during the third semester of *High Noon*. Instead of waiting on assets and placing them slowly over the course of the semester, Level Designers were instead given one asset to act as a placeholder until large quantities of assets were completed. This one asset was a simple cube, textured with a black-and-white checkerboard pattern, and was to be used as a representation of every planned asset in each level. Level Designers built entire levels out of this one asset, scaling and rotating where necessary, as well as building complex shapes from multiple instances of the cube.

The results were staggering. All Level Designers used the cube asset with great success, accurately illustrating complex elements such as trees, terrain, and architecture with ease. Within weeks we had working visual prototypes of all of our *High Noon* levels, and were able to soon after conduct preliminary playtesting. As a result of redesigning the level design pipeline, we were able to send our levels early through multiple iterations of playtesting, as well as able to visualize what the final levels would look like without any completed visual assets for the game. The instant visual feedback on our levels allowed us to make fine adjustments and spot design flaws early in production, saving us a large amount of time that was lost during the first two semesters of *High Noon*. The process was so successful that it was also used in the production of *D.A.V.I.S.*
Portal was a widely successful PC game released by Valve Corporation in 2007 that featured the use of weighted cubes to solve environmental puzzles and shield the player from damage. The “Companion Cube” was a unique version of the weighted cube, designed to provide the player with incentive (via “companionship”) to carry it with them throughout a particular series of puzzles. Both Portal and the humorous “Companion Cube” were extremely popular during the development of High Noon, and resulted in this method being affectionately referred to as “Designing by Companion Cube.”

Cautions for New Directors
New Chico State Game Studios Directors are given a mountain of responsibility when they are assigned to a new project. Scheduling, personnel issues, designing the game, creating documentation, and learning about any new technologies are just some of the many diverse problems for which the new Director must find solutions. With so much work moving so quickly, it is easy to make oversights that have large impacts on the project if left for too long. Below are a few notable cautions for new Directors based off of the more significant problems that High Noon faced during production.

The Problem with Design Documents
Design Documents are a subject of much debate, as no standard processes exist in the game development industry for documentation of any kind. Most companies, in fact, closely guard their production processes and documentation standards. Bethesda Softworks, Quantic Dream, and Cyan Worlds, for example, were contacted during the
production of *High Noon* with inquiries about their standards for game documentation and design document creation, and all companies declined to share their processes for confidentiality reasons.

“The lack of standards in writing good game design documentation has resulted in most designers and design teams shooting from the hip, throwing everything but the kitchen sink into a game design document” (Schubert, 2008). Indeed, design documents are difficult to write because games are usually very complex, multifaceted, interdisciplinary productions. It is the general inclination of the game designer, therefore, to make the game design document an all-inclusive, complete reference for the game, covering everything from back story to technical specifications for game elements. The design document for *High Noon* was written in a similar fashion, with the final document approaching thirty-two pages and serving as the primary means of disseminating information about the game to the team.

The *High Noon* design document was considered to be a living document and was periodically updated throughout the production of *High Noon* as changes were made to the design and back story. While at first it was an extremely effective means of organizing the elements of the game needed for implementation, it quickly became a cumbersome and ineffective means of communication with the team. Updates occurred slowly because of the massive size of the document, and it became increasingly difficult for team members to identify the sometimes subtle changes in the document after their initial read. Programmers found the document particularly unhelpful; it lacked vital information they needed for design or was organized in such a way that it was difficult to develop code-based design specifications. In the end, the *High Noon* design document’s
greatest use was to bring new talent onto the project, which it did very well, but it was generally never that helpful to the team as a whole.

The design document’s failings were widely because of the manner in which it was written. Intended to encompass all of the information that all teams would need to create *High Noon*, it quickly became bloated and difficult to navigate, essentially obscuring more information than it clarified. The design document for *High Noon* catered well to the sub-teams that required back story and art direction, but generally failed to deliver information in a logical manner to the Programming and Level Design teams. “In order to convey design decisions, you have to consider the audience that you are writing for,” says Scott Hackett in his article, *How to Write an Effective Design Document*, “[T]here is a need for two design documents . . .[and] each document serves a different and equally valuable purpose,” (Hackett, 2007). Because *High Noon*’s design document was not expressly written with Programmers and Level Designers in mind, the information that they needed to implement the game was not easily accessible or missing entirely. Conversely, had the expressly technical design specifications been included in the original design document, it is possible that miscommunications with the teams focusing on art and visual design might have occurred. Hackett’s ideas regarding the creation of two design documents tailored to the needs of the types of sub-teams reading them is certainly an approach that *High Noon* may have benefited from.

Additionally, some game companies have adopted internal wikis as a means of delivering concise, organized information to different sub-teams on a development project. A *wiki* is “a type of web page designed so that its content can be edited by anyone who accesses it, using a simplified markup language,” (Oxford English
Dictionary, 2007). A wiki would not only render the design document searchable, but easily broken into sections for specific teams. Updates could be easily tracked and accessed, minimizing confusion on revisions and late additions to the design document. Moreover, the design document could become a collaborative effort instead of the sole responsibility of the Director, which could increase the rate at which the document is updated, as well as minimize errors and inconsistencies. Information could be delivered in a specialized manner, ensuring that teams receive the details pertinent to completing their specific tasks. It is highly recommended that delivering design documents by wiki be explored in future projects.

Switching to a “Next-Gen” Engine

*High Noon* is the only game produced by Chico State Game Studios that has incurred a mid-production engine change. During the first two semesters of production, *High Noon* was made using Epic’s *Unreal 2004* game engine through the *Unreal Tournament 2004* editor. At the time of production, the *Unreal 2004* engine was considered to be on the low edge of current generation technology.

At the end of *High Noon*’s second semester of production, Chico State Game Studios decided to shift to a “next-gen” engine, allowing for higher detail game environments and more sophisticated functionality. Upgrading to Epic’s *Unreal 3* engine was a logical choice, as Chico State Game Studios already had three years experience with the *Unreal 2004* engine, and commonalities existed between the two engines. Originally, it was intended that *High Noon* be “ported” to the *Unreal 3* engine as a means of easing into the new next-generation technology. Porting is “the adaptation of a piece of software for a different type of machine or system” (Oxford English Dictionary, 2006),
and is often used when creating console video games so that one game may be played on multiple gaming systems.

Research indicated that the processing of porting would be a simple one; it was expected that most assets generated for the *Unreal 2004* version of *High Noon* would be easily imported into *Unreal 3*, and that levels could be converted directly from *Unreal 2004* into new, *Unreal 3* levels. In terms of programming, *Unreal 3* used an updated version of *UnrealScript*, so it was thought that most of our code would move easily to the new engine. The port was projected to be easy to assemble, providing us with an excellent opportunity to familiarize ourselves with the new engine before the next Chico State Game Studios project started the following semester.

In reality, none of the tools that we intended to use to ease the burden of porting *High Noon* worked with any accuracy or reliability, and we found ourselves faced with rebuilding all of the major elements of the game from scratch for the new engine. Old art assets designed for *Unreal 2004* looked flat and simplistic in the new engine and required significant updates. Levels had to be completely rebuilt due to massive errors with the level conversion software. Significant changes to *UnrealScript* meant that much of our code had to be restructured and redesigned for the new engine. Essentially, we found ourselves tasked with learning a new game engine, establishing new policies and pipelines to reflect changes in the engine, recreating all of our old art assets, rewriting our code, and re-designing all of our levels in one semester’s time. The “port,” we learned, would not be a port at all, but a complete redesign and reimplementation of *High Noon* in a larger, more complicated engine, expected to be complete in one-half the time it took to make the *Unreal 2004* version of the game. Though the third semester of *High Noon*
quite naturally did not lead to a complete, functioning game, *High Noon*’s third semester greatly succeeded as a means of establishing a solid knowledge base for the new *Unreal 3* engine and designing a solid pipeline for next-generation game development.

For a student studio such as Chico State Game Studios it is advised that engine switching or upgrading occur as seldom as possible. Mastering industry-standard game engines takes years of research and experience, and frequent engine switching does not allow for the level of familiarity necessary for the production of a sophisticated game in a current generation engine. Chico State Game Studios worked with the *Unreal 2004* engine for three years and still lacked the ability to quickly implement even simple game features during the production of *High Noon*. Expertise is valuable and takes time to develop with any complicated technology. It is highly recommended that Chico State Game Studios stay with a chosen engine for as long as possible to allow for the development of a strong knowledge base.

It may be argued that using a next-generation engine is more valuable to students because it reflects more closely the tools being currently used in industry. However, choosing to use a new engine, and especially a next-generation engine, should be a calculated move on the part of the student game studio. Choosing to move to a new engine too early in the engine’s life, for example, could cause problems with finding proper documentation on how to use the engine’s features. Even the plentiful resources on the internet for *Unreal 3* were riddled with holes and inconsistencies, making development overall very difficult. In summary, not only should the team be prepared for a move to a new engine, but it should first be verified that proper documentation exists to implement the desired features in the game before the move occurs.
In addition, with next-generation technology comes next-generation file sizes. The *Unreal 2004* version of *High Noon* created during the first and second semesters of production reached about 400 megabytes in size in the final “alpha” version of the game. By the end of the third semester of production, when *High Noon* had been recreated in the *Unreal 3* engine, the game quickly bloated to around 2 gigabytes, more than quadrupling in size. File size and the number of large files necessary to complete a next-generation title should be carefully considered before switching to a next-generation engine, and carefully monitored during production to ensure the game does not bloat to an unmanageable size.

The H.Y.D.E. Dilemma

One of *High Noon*’s greatest limitations was its weak preproduction. Completed during the summer before the first semester of *High Noon*’s official production, preproduction consisted primarily of solitary brainstorming by the Director based loosely off of the rough treatment of the selected game idea pitched the semester prior. Chico State Game Studios lacked any definite preproduction standards at the time, leaving the Director to their own devices in creating the production plan for the year-long project.

The manner in which *High Noon* was produced was largely a series of experiments: “what-if” scenarios in combination with observations of what the team and project were lacking inspired most of the pipelines that were tested during the course of the project. One of these experiments was the corralled approach to game design that was explored during the creation of the original game design document for the game. *High Noon* was structured with the intention of giving the team wide creative control over the
direction and content of the game. With that in mind, *High Noon*’s original design
document was primarily suggestive, containing a great deal of passive voice and rough
ideas, offered to the team as a “corral” in which to develop new ideas as an effort to
impart creative ownership to the Chico State Game Studios team.

One of the ideas roughly explored in the *High Noon* design document was the
dual personality of the main character, Jekyll. According to the *High Noon* mythology,
Jekyll was a member of Sutter’s Mill, a rogue establishment of scientists and rebels
seeking to remain independent from the increasing number of cities and territories
controlled by a few wealthy and corrupt land barons. One such land baron, Boz Novak,
sought to control Sutter’s Mill, sending in waves of attacks in hopes of weakening its
defenses. Frustrated with his lack of success, Novak turns to parasitic warfare, capturing
Jekyll and infecting him with the H.Y.D.E. parasite: an infectious, heavily contagious
parasite designed to take Sutter’s Mill down from the inside. Novak then returns Jekyll to
Sutter’s Mill, hoping that they will blindly let him into the settlement, and allowing his
parasite to spread. Jekyll’s infection serves as one of the core gameplay elements of *High
Noon*: by letting the H.Y.D.E. parasite take over, Jekyll is able to jump higher, use
specialized weapons, and gains a significant health and defense bonus. It was hoped
during the publication of the original game design document that this relationship would
introduce a strategy element into the game, encouraging the player to plan when they use
either persona to navigate the enemies and environments.

The Jekyll and H.Y.D.E. relationship is central to *High Noon*’s mythology,
functionality, and player motivation, yet it was never formally tested for functionality.
Because the design document was intended for suggestions to the team, the
Jekyll/H.Y.D.E. dynamic was never fully explored or developed after the game went into full production. The demands of production became so intensive, in fact, that the design details of the Jekyll/H.Y.D.E. relationship were frequently lost or misinterpreted. Sometimes, Jekyll could run very quickly when he activated his H.Y.D.E. powers; other times, it was decided that decreased speed would be more appropriate. Level Designers often forgot to consider the Jekyll/H.Y.D.E. dynamic when designing quests and puzzles for the game, or had trouble designing appropriately balanced puzzles for the ever-changing specifications of the effects of the H.Y.D.E. parasite.

By the time the game was ported to Unreal 3 in its third semester of production, the fact that the H.Y.D.E. parasite had not been formally designed and documented fell to the wayside in favor of learning the new Unreal 3 technology and exploring what tasks would be necessary to creating a successful port. It was realized very late in the project that this element that was once key to the game’s back story and character motivation was now one of its biggest design obstacles; it had been forgotten about for so long that most of the levels no longer catered to H.Y.D.E. at all, or when they did, the best we could think to do were clichéd game standbys like jumping puzzles that are notorious for frustrating the player. However, by the time we realized that H.Y.D.E. no longer had a place in the High Noon gameplay it was already so ingrained into the image of the game that it was near impossible to change. Jekyll’s motivation for pursuing Novak was related to the H.Y.D.E. parasite, character dialog hinged upon the idea that Jekyll was infected, most art generated by the project featured the creation of the parasite and Jekyll turning into H.Y.D.E., and even the logo generated for the project featured the two faces of Jekyll and H.Y.D.E. There was truly no distancing the project
from its malfunctioning core dynamic, and as a result the fun of the game was greatly compromised.

*High Noon* had essentially no design team before it went into full production. According to precedent, it was the responsibility of the Director to complete the design of the game prior to production, and often this task was completed by the Director alone. During this time, game designs should have been heavily tested for balance and functionality. Instead, *High Noon* was largely designed on the fly, unable to change quickly, or sometimes at all, when problems like the H.Y.D.E. parasite occurred. Additionally, it is extremely bad game design practice to greenlight a project that has not been tested at its core for functionality and fun. In short, *High Noon* was in many ways not ready to enter production when it did, and major problems that existed at the beginning of the production pipeline haunted the project throughout all of production.

Strong preproduction is vital to the success of a video game production. Projects, especially ones of the size and scale that Chico State Game Studios typically undertakes, become more and more rigid the longer they are in production. Mistakes like the H.Y.D.E. parasite create vast problems for a project, and possess great potential for crippling a project beyond recovery.

**Conclusion**

The accomplishments of the past year and a half for Chico State Game Studios have been significant. Not only has the organization developed functioning, close-to-industry production pipelines and team structures, but it has also embraced its foundations in academia and created production standards appropriate for a
classroom/game studio hybrid. Students in Chico State Game Studios classes are more prepared for industry than ever, and there is a growing external interest in the level of sophisticated production that Chico State Game Studios is achieving.

Chico State Game Studios is a unique organization that offers students a once-in-a-lifetime opportunity to work in a large, unified team toward the creation of a complex, year-long development project in a safe, academic environment. The skills and techniques students take away from a Chico State Game Studios project are directly applicable to those used in the video game industry. This project reinforces the foundation on which Chico State Game Studios procedures were developed, and ensures that future generations of Chico State Game Studios projects have a solid understanding of the complex and delicate processes that are used to make next-generation video games.

The documentation presented in Chapter III of this project write-up provides a detailed look into the creation of a student game project from original concept to final product, as well as creates a precedent for documentation and archiving of knowledge for future Chico State Game Studios Directors. It is my hope that this project will help eliminate the lost knowledge and experience of prior Chico State Game Studios projects, and instead start a new standard of support and forethought to future productions.

Additionally, Chico State Game Studios is responsible for creating a complex, sophisticated, fun game, but it is also responsible for challenging and educating team members on standard game production practices and techniques. It is my hope that the flexible production methods outlined in Chapter III of this paper will act as a framework for future Chico State Game Studios projects, as well as remind incoming Directors of
the importance of the academic and educational facets of a Chico State Game Studios production.

This project has comprehensively documented one of the most transitional and positive stages of Chico State Game Studios’ transformation into a successful student game studio. The policies developed during the life of the *High Noon* and *D.A.V.I.S.* projects described in this paper will provide a vital framework for projects to come, fostering a knowledgeable, empowered organization of student game developers who themselves will contribute to the continued refinement of future game pipelines.

For additional information on this project, source files, or any other supplemental materials, please visit the author’s website at http://www.alishathayer.com.
REFERENCES
REFERENCES


http://www.cyan.com/company/index.php


http://www.gamasutra.com/features/20060424/hawkins_01.shtml


http://dictionary.oed.com.mantis.csuchico.edu/cgi/display/30007244?keytype=ref&ijkey=u.LmPeIBr409A


http://dictionary.oed.com.mantis.csuchico.edu/cgi/display/50293088?keytype=ref&ijkey=cLkkiOsjzUgJ6


APPENDIX A
RECOMMENDED READING


DEFINITION OF TERMS

1. **ActorX Plugin** – A plug-in for most major modeling packages that allows the exportation of skeletal and static meshes for use in the *Unreal* engine.

2. **Application Programming Interface (API)** – The classes, functions, and variables in a given programming language.

3. **Binary Space Partitioning (BSP) Geometry** – The most primitive type of geometry supported by the *Unreal* engine, composed primarily of cube-derived shapes with complex surface textures. BSP geometry is either subtractive or additive, and therefore either adds to a level composed of negative space or subtracts from a level composed of positive space.

4. **Channels** – The many inputs for a surface on a 3D model. Includes diffuse, transparency, specularity, et cetera.

5. **Cinematics** – In-game movies that impart narrative to the player.

6. **Clipping** – Traversing a 2D plane in 3D space. For this thesis, clipping generally refers to the undesired effects resulting from 3D models that pass through one another rather than behaving as solid objects.

7. **Death Match Game Type** – Popular in first-person shooter (FPS) games, the Death Match game type is a competition between many players to get the most player-kills.

8. **Decals** – A 2D image plane that rests over a textured 3D object in a game environment. Used primarily to add levels of unique detail to objects that are frequently replicated in a level.

9. **Diffuse Texture** – Also referred to as a color map, a diffuse texture provides flat color textural details through the application of a bitmapped image to a 3D surface.

10. **Dots Per Inch (DPI)** – The number of pixels present per inch on a monitor.
11. **Dramatic Elements** – Game elements that “engage players emotionally with the game experience . . . like premise, character, and story” (Fullerton, 2008, p. 81).

12. **Formal Elements** – Game elements “that form the structure of a game. Without them, games cease to be games . . . Players, objective, procedures, rules, resources, conflict, boundaries, and outcome” (Fullerton, 2008, p. 43).

13. **Game Assets** – Any digital element created for a game, including: code, animations, 3D models, concept art, textures, and levels.

14. **Game Conversion** – The act of moving an element or all of a game from its originally designed framework to another. See also, “Game Port.”

15. **Game Design Document** – A document containing all of the information needed to create the game. This includes: game background, target audience, intended platforms, system requirements, detailed descriptions of gameplay, game features, detailed level/environment descriptions, major development tasks, and an estimated schedule.

16. **Game Engine** – A piece of software that controls systems and assets necessary for the creation and interaction with video games.

17. **Game Port** – “The process of adapting software so that an executable program can be created for a computing environment that is different from the one for which it was originally designed” (Porting, 2008).

18. **Geometry (3D Modeling)** – The make-up of polygons that composes a 3D model.

19. **Head-Up Display (HUD)** – A transparent overlay that is present on a game screen. HUDs are typically used for displaying game data such as health, ammunition, mini-maps, and status messages.

20. **High-Level Concept** – An overview of all game elements such as the story, characters, game flow, game uniqueness, and game systems. High-level concepts generally do not describe how elements are to be implemented.

21. **Level Paintovers** – The act of taking a screenshot of a barren in-game level and passing it to a Concept Designer to add new architecture, props, and lighting. Used to assist Level Designers in the aesthetic design of their levels.
22. **Low-Level Concept** – A concept containing the technical details of a game element’s implementation. Whereas the high-level concept explains that elements exist, the low-level concept explains how they exist.

23. **Maps : Alpha** – An alpha map controls the transparency of a texture through the application of a black and white bitmapped image. Generally, black elements on an alpha map are transparent, while white elements are opaque.

24. **Maps : Normal** – A map composed of red, blue, and green values that controls the 3-dimensional bump of a surface on a 3D model.

25. **Maps : Reflection** – A map that controls what is reflected in a reflective surface.

26. **Maps : Specularity** – A map that controls the specular reflectivity of a surface.

27. **Model Sheets** – Drawings generated by a Concept Design team that explain in detail the artistic requirements for the creation of a 3D model, including: scale, texture, shape, and context.


29. **Modeling Package** – A piece of software that enables a digital artist to create 3D models, such as *Newtek Lightwave*, *Autodesk Maya*, or *Autodesk 3D Studio Max*.

30. **Next-Generation (Next-Gen)** – Games that use advanced computer graphics techniques to create high-resolution, technically complex games.

31. **Paper Doll System** – An in-game system that allows for the quick interchanging of costume elements on a player’s avatar, or in-game character.

32. **Paper Prototyping** – The act of testing game designs on paper before implementing them electronically. Paper prototyping allows for the testing of core game functionality and identification of design flaws early in a game’s production.

33. **Playtesting** – The act of playing a game for the purpose of identifying errors or bugs.

34. **Polygon (3D Modeling)** – The three-dimensional shape resulting from the combination of edges, vertices, and surface normals in a 3D modeling package.
35. **Production Pipelines** – A definition of rules and procedures for the development of video games that outline dependencies, possible pitfalls, milestones, and contingency plans.

36. **Ragdoll Physics** – “A type of procedural animation that is often used as a replacement for traditional static death animations” (Ragdoll Physics, 2009).

37. **Rigging** – The process of building a digital skeleton on which to deform a 3D mesh in computer animation.

38. **Steampunk** – A genre of fiction that features steam-powered technological innovations usually in the context of the Victorian era or other eras where computing and advanced technology do not belong.

39. **Texturing** – The act of creating and applying the painted surface of a 3D model.

40. **UV Stretching** – An undesired effect of improperly creating UV maps. UV stretching results in applied texture maps incorrectly appearing on a 3D model and instead looking stretched.

41. **UV Mapping** – The act of laying out the polygons composing a 3D model onto a 2-dimensional space, much like the act of flattening a 3D image of an Earth onto a 2D surface. UV maps are 2D representations of 3D models.

42. **UV Snapshots** – Screen-captures or modeling package-generated images that show the 2D representation of a 3D model.

43. **UV Texturing** – The act of UV mapping a 3D model and creating a bitmap texture that corresponds to the 2D placement of geometry.

44. **Vertex (3D Modeling)** – A corner point on a polygon.

45. **Weeklies** – Meetings that occur once weekly to review the week’s work and set goals for the following week.
HIGH NOON DESIGN DOCUMENT

HIGH NOON

CHICO STATE

GAME STUDIOS

Design Document v. 1.6

Alisha Thayer
Director
CSGS Fall 2008
Table of Contents

REVISION HISTORY ........................................ 4

I. OVERVIEW/BACKGROUND ................................ 6
   BACKGROUND ........................................ 6
   OVERVIEW ........................................... 6
   TONE .................................................. 8
   WHAT IS STEAMPUNK? .................................. 8

II. TARGET AUDIENCE ...................................... 9

III. PLATFORM .............................................. 9

IV. SYSTEM REQUIREMENTS ............................... 9

V. GAMEPLAY ............................................... 10
   OVERVIEW ........................................... 10
   CONTROLS ............................................ 10
   INTERFACES ......................................... 10
   RULES .................................................. 10
   SCORING/WINNING CONDITIONS ..................... 11

VI. WEAPONS ............................................... 11
   THE SCHMITTOUT SMASHER PIPE WRiNCHE R (JEKYLL) 11
   RED SHIRT’S REVENGE™ RAY GUN (JEKYLL) ............. 12
   RELOAD-No-MORE™ AUTOMATIC PISTOL (JEKYLL) ....... 12
   DR. McNAMARA’S TRIPLE-SHOT SCOPEd lRiFLE (JEKYLL) 13
   THE LOCK ROCKET Mini ROCKET Handgun (JEKYLL) .... 13
   THE SPIKE-EYED RIaIL SPOKE GUN (JEKYLL) ........... 14
   THE RX55 PORTaTABLE TURRET (HYDE) ................ 14
   THE EXPLODING™ EXPLODING CANNONBALL LAUNCHER (HYDE) 15
   HYDE’S FISTS (HYDE) .................................. 16

VII. LEVELS .................................................. 16
   SUTTER’S MILL: ABOVE GROUND ...................... 16
   RECONSTRUCTED CHURCH ................................ 18
   PRE-Collapse GRAVEYARD (ST. MARY’S CEMETERY) .... 19
   THE ABANDONED TRAIN DEPOT ......................... 20

VIII. FLOWCHART ........................................... 21

IX. GAME CHARACTERS ..................................... 21
   CHARACTER DESIGN .................................... 21
   THE PLAYER-CHARACTER (JEKYLL/HYDE) ............... 21
   THE PRIMARY ANTAGONIST: BOB NOVAK ................. 22
   MONSTERS AND ENEMIES ................................ 22
   Mechanical Spiders .................................. 22
   Mechanical Widows ................................... 23
   Steam-Powered Dragonflies ............................ 23
   Armored Dragonflies ................................ 24
   Caterpillar .......................................... 24
   Lightning Bugs ..................................... 25
   Grasshopper ......................................... 25
   FRIENDS AND ALLIES ................................... 26

X. STORY ..................................................... 26
   COMPLETE STORY (SUMMARIZED) ...................... 26

High Noon Design Document v. FINAL
© 2007/2008 Chico State Game Studios

Page 2 of 31
| BACKSTORY ........................................................................ | 26 |
| NARRATIVE DEVICES .......................................................... | 27 |

| XI. CHARACTER BIOGRAPHIES ................................................. | 28 |
| BOB NOVAK ..................................................................... | 28 |
| Jekyll (Player Character) .................................................. | 28 |
| ANNE GORDON ............................................................... | 29 |
| Dr. Stephen Hudson ........................................................ | 29 |
| JENNIFER WATTS ............................................................ | 30 |

| XII. CUSTOM HUD ............................................................ | 31 |
| PROPOSED ELEMENTS ........................................................ | 31 |
## Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 August 2007</td>
<td>- Initial document creation</td>
</tr>
<tr>
<td>06 September 2007</td>
<td>- Added “Character Biographies” section.</td>
</tr>
<tr>
<td></td>
<td>- Added Level Objectives, Available Weapons, and Enemy Types sections to Sutter’s Mill Level.</td>
</tr>
<tr>
<td></td>
<td>- Updated Friends and Allies section.</td>
</tr>
<tr>
<td></td>
<td>- Re-arranged weapons by level of potential destruction.</td>
</tr>
<tr>
<td></td>
<td>- Polished Overview/Background section.</td>
</tr>
<tr>
<td></td>
<td>- Added potential enemy types.</td>
</tr>
<tr>
<td></td>
<td>- Added section on custom HUD.</td>
</tr>
<tr>
<td></td>
<td>- Removed “Humorously Steam-Powered Enemies” from the Enemy Type list.</td>
</tr>
<tr>
<td>10 September 2007</td>
<td>- Added info for each weapon type: rounds, maximum ammo, destructive power (on a 10 point scale), ammo type, secondary, and secondary max ammo (if applicable).</td>
</tr>
<tr>
<td></td>
<td>- Fixed the embarrassing typos in the “Character Biographies” section.</td>
</tr>
<tr>
<td></td>
<td>- Added a melee weapon, “Jekyll’s Pipe Wrench” to the Weapon’s section.</td>
</tr>
<tr>
<td></td>
<td>- Added a “Main Plot Goals” section to the Sutter’s Mill, Reconstructed Church, Graveyard, and Derelict Train level descriptions</td>
</tr>
<tr>
<td></td>
<td>- Introduced the “Grasshopper” enemy type</td>
</tr>
<tr>
<td></td>
<td>- Introduced the “Lightning Bug” enemy type</td>
</tr>
<tr>
<td></td>
<td>- Updated Pre-Collapse Graveyard description.</td>
</tr>
<tr>
<td></td>
<td>- Updated Derelict Train section.</td>
</tr>
<tr>
<td></td>
<td>- Added a detail about Hyde’s ability to jump significantly higher than Jekyll in the Player Character (Jekyll/Hyde) section of part 10</td>
</tr>
<tr>
<td></td>
<td>- Added a steam-powered arm to Novak’s character description. It can even produce a protective force field! Neat!</td>
</tr>
<tr>
<td>07 October 2007</td>
<td>- Added weapon concepts to weapon descriptions.</td>
</tr>
<tr>
<td></td>
<td>- Fixed Explosinator description; yes, it’s a rocket launcher, not a grenade launcher. Oops.</td>
</tr>
<tr>
<td></td>
<td>- Added statistics to all enemies.</td>
</tr>
<tr>
<td></td>
<td>- Added pictures and descriptions for the Lightning Bug and the Grasshopper.</td>
</tr>
<tr>
<td></td>
<td>- Removed “On Hold” status from the Hyde Parasite Soldiers. We will be conceptualizing them this semester with the possibility of modeling/implementing them.</td>
</tr>
</tbody>
</table>
- Added "Animation Notes" and "Technical Details" to the weapons, generally changed the formatting.
- Replaced the HUD graphic with a newer, in-progress graphic.
- Upped the ammo on the Portable Turret to 70 in the chamber, 420 max ammo.
- Changed the alt-fire on the Lock Rocket to fire both rockets at once.
- Changed the ammo in-cartridge to 10 in the Rail Spike Gun.

**25 January 2008**
- Updated general overview.
- Applied an "on-hold" status to the Ultra Stylish Brass Night-vision Goggles.
- Modified level descriptions.
- Modified enemy descriptions.
- Removed a great deal of future tense from the character section that deals with concept design.
- Added renders of in-game enemies.
- Replaced old HUD with new HUD. Yay!

**28 August 2008**
- Document updated to reflect the decision to convert to the Unreal 3 engine.
- Example concept art replaced with concept art created explicitly for High Noon.
- Refined Level Descriptions
- Removed the BB Gun from the weapon arsenal.
- Removed the Golden Gun from the weapon arsenal.
- Removed the Divine Trumpet from the weapon arsenal.
- Removed the Steampunk Goggles from the weapon arsenal.

**2 September 2008**
- Modified the "Characters" section, added pictures.
- Modified level descriptions.

**2 October 2008**
- Added re-balanced weapon information
- Added re-balanced enemy information
- Added Hyde's Hands as a Hyde weapon type
- Modified the Jekyll/Hyde relationship in such a way that Hyde no longer gets increased speed, but rather decreased speed.
- Removed the Rolly Pollies (explosive and standard) from the HN enemy lineup.
- Modified Boz Novak's description to include the changes to his arm (being able to summon enemies and shoot), and removing the force field.
1. Overview/Background

High Noon is a single-player action FPS set in a Steampunk-flavored wild west, created using the Unreal 3 engine, and based on a working prototype built in the Unreal 2.5 (Unreal Tournament 2004) engine.

Background

After the Great Collapse, the United States crumbled into a land of warring city-states, renegade law enforcement, and ruthless, localized governments. Cities deteriorated as residents fled for safe-havens and settlements that promised protection. Some settlements became hideouts for scientists, academics, families, and others not suited for vigilante life in the wastelands. These settlements were usually well-hidden and heavily guarded; many stretching deep underground for maximum protection, and often hidden within the wreckage of former large cities. Most noteworthy of these settlements is Sutter’s Mill, a sprawling refuge in the former Sacramento Valley.

Settlements like Sutter’s Mill have become heavily sought after by a growing population of land barons, who, in the absence of a unified government, are ruthlessly usurping land to turn a profit from those seeking some semblance of civilization. The existing settlements are razed; scientists, technologists, engineers, and those with skills deemed of worth are captured and recruited, strengthening the land baron forces and making them a formidable power.

At the forefront of the land barons seeking to raze Sutter’s Mill is Boz Novak, a powerful baron who has managed to collect a large portion of the land in the mid-west. Seeking to stretch his influence to the former California coast, Novak has spent the last three years sending wave after fruitless wave of attacks on Sutter’s Mill’s formidable exterior defenses. Novak’s continued failures have caused him to turn to more drastic measures: parasitic warfare.

Using a recently discovered human parasite that causes severe episodes of rage and increased strength, Novak “enhanced” a percentage of his army to form a brute-force branch. As an added precaution, Novak managed to obtain and infect a member of the Sutter’s Mill settlement in an effort to destroy the colony from the inside.

Overview

High Noon takes place immediately after the captured citizen (the player) is returned to Sutter’s Mill. Though unwilling to allow the player to return to the depths of the colony due to the infectious qualities of the parasite, the citizens of Sutter’s Mill have developed a plan to utilize the player’s infection to their advantage by turning him/her back on Novak’s army.

High Noon Design Document v. FINAL
© 2007/2008 Chico State Game Studios
Page 6 of 31
High Noon’s most unique quality is the Jekyll/Hyde dynamic of having a character infected with an increasingly influential parasite. Serum pickups are the primary factor in scarcity and strategy for the majority of the game as they are the catalyst in awaking the Hyde qualities of the parasite. When the character is Hyde, he experiences greater strength, higher defense, decreased speed, and the ability to use special weapons that are unavailable to Jekyll. Hyde’s reduced dexterity renders him incapable of using weapons in Jekyll’s arsenal.

Balance is key: obstacles and puzzles will lead to needing one persona over another, and the number of times that Hyde is accessible is limited to the number of serum vials that the player has available. Hyde is inarticulate and brutish, Jekyll must be used to communicate with Sutter’s Mill to advance the plot and discover new mission objectives. Allies do not respond to Hyde for fear that he will turn on the colony, and are hostile to him where they would not be to Jekyll.

High Noon takes place among the wreckage of a collapsed nation, lawless, without government, and reminiscent of the actual “wild west” of the 1800’s. Homebrew “Steampunk” flavored technology and settings are prevalent, giving the environments a unique combination of old and “new” technology: woods and brasses mingled with wires and 20th century computers.


The color palette for High Noon is composed of oranges, pinks, reds, browns, purples, and sepia tones.
Characters in High Noon have exaggerated and simplified features, though are still highly expressive and reasonably complex. Textures are rich in color though smooth in consistency. There are three visible humanoid characters in High Noon: Jekyll, the hero, Hyde, the hero’s alter-ego, and the antagonist, Boz Novak.

![Boz Novak, founder of New El Dorado and villain in High Noon. Illustrated by Ayla Richards.](image)

High Noon will be accented by western-flavored music, tinted with a subtle technological flair.

**Tone**

High Noon’s tone is primarily that of action/drama, but sprinkled with tongue-in-cheek humor and pop-culture references. Like the art direction, the tone of High Noon is vibrant, high-contrast, and over-the-top. It is an amalgam of traditional Westerns, Steampunk flair, and modern geek humor. While the primary objective for High Noon is to communicate an interesting, compelling story, it’s always nice to have a laugh or two along the way.

**What is Steampunk?**

Adapted from [http://www.brassgoggles.co.uk](http://www.brassgoggles.co.uk)

Steampunk is a genre of fiction typically set somewhere in the 1800’s during the Victorian Era (please note that High Noon is intentionally timeless). What makes...
Steampunk fantastic is that it introduces an element of technology into the time period that is obviously completely out of place. Examples include steam robots, flying castles, under-water bases, moon rockets, time machines etc. Basically, the sky is the limit when designing anything with a Steampunk flair. Throw historical accuracy to the wind!

Steampunk is heavily defined by its look: brass, dark woods, intricate craftsmanship, and other elements that in essence feel Victorian, but have something so technologically advanced about them that they could not possibly be Victorian.

Steampunk is closely related to (if not the same thing as): Victorian Science Fiction, Gaslamp Fantasy, Steampulp, Fireside Science Fiction, Neo-Victoriana, Wild/Weird West, Voyages Extraordinaire, and Scientific Romance.

For more information about Steampunk, check out the Wikipedia Steampunk entry or the Steampunkopedia at http://steampunk.republika.pl/arch/opedia.html.

II. Target Audience

The target audience for High Noon is players who enjoy a high-action FPS with strategy elements. Levels are designed in such a manner to encourage exploration, so exploratory players will enjoy hidden rewards. Players who enjoy unique weapon styles will enjoy High Noon’s creative twist on traditional FPS weapon types.

III. Platform

High Noon is designed to run on Windows-based PC’s per the requirements of the Unreal 3 engine.

IV. System Requirements

High Noon’s system requirements match those of Unreal 3, which are:

<table>
<thead>
<tr>
<th>Minimum System Requirements</th>
<th>Recommended System Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows XP SP2 or Windows Vista</td>
<td>2.4+ GHZ Dual Core Processor</td>
</tr>
<tr>
<td>2.0+ GHZ Single Core Processor</td>
<td>1 GBytes of System RAM</td>
</tr>
<tr>
<td>512 Mbytes of System RAM</td>
<td>NVIDIA 7800GTX+ or ATI x1300+</td>
</tr>
<tr>
<td>NVIDIA 6200+ or ATI Radeon 9600+</td>
<td>Video Card</td>
</tr>
<tr>
<td>Video Card</td>
<td>8 GB of Free Hard Drive Space</td>
</tr>
<tr>
<td>8 GB of Free Hard Drive Space</td>
<td></td>
</tr>
</tbody>
</table>
V. Gameplay

Overview

High Noon is a fast-paced action FPS heavily influenced by Unreal Tournament 3 in terms of controls and navigation. High Noon implements the standard “WASD” + mouse FPS control methods, 1-9 key weapon selection, et cetera.

Gameplay is moderately influenced on whether the player is in Jekyll or Hyde states, as different weapon types are available to the two personas. Most notably, weapons that are available to Jekyll are not available to Hyde. Allies will not interact with Hyde, and even go hostile on him out of fear, making Jekyll the primary means of communication with Sutter’s Mill.

Switching from Jekyll to Hyde is achieved by activating a one-use serum pickup (Hyde Catalyst Serum, HCS) that can be found in-game. The player may only carry two serum vials at a time, making it necessary to plan when it is most appropriate to trigger the Hyde parasite.

Returning to Jekyll can be achieved by selecting to initiate the transformation from Hyde. The player must struggle with Hyde (by pressing the left and right keys in rapid succession, eventually filling a progress bar), and when the struggle is complete, the player returns to Jekyll. As the parasite grows stronger, this process may take more time.

High Noon employs a number of other FPS standbys such as a HUD, life meter, limited ammunition, and health/weapon/ammo pickups.

Controls

Special controls in High Noon include:

<KKEY> Administer Hyde Catalyst Serum (HCS)
<KKEY> Initiate Return to Jekyll

Interfaces

In addition to custom game menus and a new HUD, High Noon also implements a small, custom interface for the Hyde-to-Jekyll transformation.

During transformation, a message and progress bar will appear on the screen, prompting the player to “Press left and right in rapid succession!”, during which the bar will increase. When 100% full, the player successfully returns to Jekyll’s form.

Rules

The game is over when the players hit points (HP) reaches zero.
Scoring/Winning Conditions

The game is considered completed when the player resolves the final conflict with Boz Novak during endgame.

VI. Weapons

The Schmitzou Smasher Pipe Wrench (Jekyll)

Mo Deled by Andrew Holifield, Textured by Alisha Thyner

An engineer's best friends are his tools. Good for times when you're in a bind, this pipe wrench packs a pretty decent wallop considering its rather small size.

Jennifer always said it'd come in handy...

Animation Notes: Please refer to Bioshock's wrench animation for animation suggestions.

**Melee Damage Only**

<table>
<thead>
<tr>
<th>Ammo (in gun):</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammo (in pocket):</td>
<td>N/A</td>
</tr>
<tr>
<td>Primary fire:</td>
<td>Hit</td>
</tr>
<tr>
<td>Secondary fire:</td>
<td>Wind-up Hit</td>
</tr>
<tr>
<td>Rate of Fire:</td>
<td>2 Hits/second</td>
</tr>
<tr>
<td>Reload Time (primary):</td>
<td>N/A</td>
</tr>
<tr>
<td>Reload Time (secondary):</td>
<td>N/A</td>
</tr>
<tr>
<td>Primary Damage:</td>
<td>5</td>
</tr>
<tr>
<td>Secondary Damage:</td>
<td>10</td>
</tr>
<tr>
<td>Notes:</td>
<td>None</td>
</tr>
</tbody>
</table>
Red Shirt’s Revenge™ Ray Gun (Jekyll)

Ammo (in gun): 25
Ammo (in pocket): 125
Primary fire: Fire (laser beam)
Secondary fire: Charged Shot
Rate of Fire: 8 shots/second
Reload Time (primary): 2 seconds
Reload Time (secondary): N/A
Primary Damage: 5
Secondary Damage: 20
Notes: Hold RMB to charge, release to fire.

Never be caught unprepared again! This highly concealable, lightweight science experiment is perfect for any gunnman, regardless of experience or expertise!

The ultra-bright green rays cause little-to-no kick, making this weapon so easy to fire that even a child could use it! Not that we’d recommend that; these beauties get hot!

Warning: Users named Ray may experience unexpected results using the Red Shirt’s Revenge™ Ray Gun.

Technical Details: The Red Shirt’s Revenge has held with one hand and is relatively small and lightweight. It is reloaded by inserting an ammo capsule into back of the gun. It has little recoil.

Animation Notes: Ideally, it would be cool if the liquid could splash in the tube as the player moves.

Reload-No-More™ Automatic Pistol (Jekyll)

Ammo (in gun): 24
Ammo (in pocket): 96
Primary fire: Fire (bullets)
Secondary fire: Canister Grenade
Primary Rate of Fire: 6 shots/second
Secondary Rate of Fire: 1 shot/2 seconds

This heavy-duty pistol packs a serious punch, firing medium-velocity rounds at high speeds.

Possessing three 8-round chambers, this gun can pump out enough lead to bring down even the toughest enemies.

Reload-No-More™: Because More is More.

Technical Details: The actual model of this gun ended up with the ammo chamber being on the bottom instead of the top. It does spin as the gun fires.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reload Time (primary)</td>
<td>.5 second</td>
</tr>
<tr>
<td>Reload Time (secondary)</td>
<td>1.5 second</td>
</tr>
<tr>
<td>Primary Damage</td>
<td>7</td>
</tr>
<tr>
<td>Secondary Damage</td>
<td>20</td>
</tr>
<tr>
<td>Notes</td>
<td>None</td>
</tr>
</tbody>
</table>

**Dr. McNamara’s Triple-Shot Scoped Rifle (Jekyll)**

Are you the type of sharp-shooter who prefers to see your enemy’s nose-hairs? Well, Dr. McNamara has the rifle for you!

With this expertly precise, highly-magnified brass beauty, you can pass time waiting for the perfect shot by counting the pores on your target’s forehead, even if they moisturize!

Dr. McNamara’s unique Triple-Shot Scoped Rifle is outfitted with three rotating barrels, enabling the user to chamber three separate rounds for extra-fast shootin’.

Go into battle looking your enemy in the eye . . . even from 500 feet away!

**Technical Details:** This weapon has three barrels and fires in a similar fashion to the portable turret in that the barrels spin. The gun doesn’t spin up like the turret, it just rotates as it shoots.

Not satisfied with the traditional gunman’s arsenal? Get a kick out of people saying, “Is that a rocket launcher in your pants, or are you just happy to see me?” Well, now you can answer, “Why yes, that IS a rocket launcher!”

The Lock Rocket Mini Rocket Handgun is for you! The sleek design is slightly larger than your average revolver, but with more kick than even a powerful shotgun!
### The Spike-Eaze Rail Spike Gun (Jekyll)

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammo (in gun)</td>
<td>8</td>
</tr>
<tr>
<td>Ammo (in pocket)</td>
<td>32</td>
</tr>
<tr>
<td>Primary fire</td>
<td>Fire (rail spike)</td>
</tr>
<tr>
<td>Secondary fire</td>
<td>Fire (all, or charged)</td>
</tr>
<tr>
<td>Primary Rate of Fire</td>
<td>1 spike/second</td>
</tr>
<tr>
<td>Secondary Rate of Fire</td>
<td>6 second charge for all 8 spikes</td>
</tr>
<tr>
<td>Reload Time (primary)</td>
<td>5 seconds</td>
</tr>
<tr>
<td>Reload Time (secondary)</td>
<td>N/A</td>
</tr>
<tr>
<td>Primary Damage</td>
<td>20</td>
</tr>
<tr>
<td>Secondary Damage</td>
<td>160</td>
</tr>
</tbody>
</table>

**Notes:** The player can release the RMB in the middle of a charge to fire the number of loaded rail-spikes at 20 damage per spike.

### The RX55 Portable Turret (Hyde)

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammo (in gun)</td>
<td>2</td>
</tr>
<tr>
<td>Ammo (in pocket)</td>
<td>10</td>
</tr>
<tr>
<td>Primary fire</td>
<td>Fire (one rocket)</td>
</tr>
<tr>
<td>Secondary fire</td>
<td>Fire (one rocket)</td>
</tr>
<tr>
<td>Rate of Fire</td>
<td>N/A</td>
</tr>
<tr>
<td>Reload Time (primary)</td>
<td>2 seconds</td>
</tr>
<tr>
<td>Reload Time (secondary)</td>
<td>2 seconds</td>
</tr>
<tr>
<td>Primary Damage</td>
<td>30</td>
</tr>
<tr>
<td>Secondary Damage</td>
<td>30</td>
</tr>
</tbody>
</table>

**Notes:**

Portability and awesome destructive power in the palm of your hand! Literally!

**Technical Details:** None, save the animation notes.

**Animation Notes:** The bottom rocket chamber moves back and forth as it fires. The hammer fires when the primary fire is activated and one rocket is launched.

Do you look at a railroad spikes and find yourself thinking about how great they'd look flying through the air at high speeds? So did we. The Spike-Eaze makes perfect use of these perfectly pointy, plentiful remnants of post-Collapse America.

Utilizing a rotating barrel, the Spike-Eaze can fire up to six shots before needing to reload. Who's working on the railroad now?

**Technical Details:** Please note that the spike cartridge is backwards in this image. I'd love to see a light emanating from the top, and the occasional burst of steam billowing from the pipe.

**Animation Notes:** This is a complicated one, animation. The wheels turn as the gun fires, as does the chamber where the spikes shoot from. Talk to me about the details.

This is post-Collapse home defense at its finest, and now it's portable!*

*With the RX55 Portable Turret you are a one man army, capable of erasing the remains of all others previously dreamed off!

**Disclaimer:** All sales final. Please consult the RX55 manual before purchase to accurately assess if the...
Ammo (in gun): 150
Ammo (in pocket): N/A
Primary fire: Fire (bullets)
Secondary fire: N/A
Rate of Fire: 10 shots/second
Reload Time (primary): N/A
Reload Time (secondary): N/A
Primary Damage: 5 per bullet
Secondary Damage: N/A
Notes: The RX-55 has become a one-shot weapon; the player may use it until it runs out of ammo and then it is dropped. Hyde may pick up RX-55 model turrets throughout the levels where the turret is available.

RX-55 is right for you.
*SUPER-HUMAN STRENGTH RECOMMENDED.

Animation Notes: The turret has a rotating barrel that spins up with as the player holds down the fire button. The barrel is spun via a crank mechanism on the side of the weapon.

The Explodinator™ Explosive Cannonball Launcher
(Hyde)

Nothing says macho like a rocket launcher, and the Explodinator is no exception! No one will question your manliness with this baby around!

Weighing in at 240 pounds, 95% of the average human population can barely lift this weapon, not to mention fire it!

Manufacturers Note: If anyone does manage to fire the Explodinator, we'd love to know! Test data is rather rare with guns like these!

Ammo (in gun): 1
Ammo (in pocket): 4
Primary fire: Fire (explosive cannonball)
Secondary fire: N/A
Rate of Fire: N/A
Reload Time (primary): 4 seconds
Reload Time (secondary): N/A
Primary Damage: 100
Secondary Damage: N/A
Notes: None

Modeled by Andrew Holifield. Textured by Althea Thayer
Hyde’s Fists (Hyde)

Like Jekyll’s wrench, Hyde’s hands provide Hyde with a means of fighting even when he’s out of ammunition.

Modeled by Scott Washington, Textured by Lindsey Anderson

**Melee Damage Only**

<table>
<thead>
<tr>
<th>Ammo (in gun):</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ammo (in pocket):</td>
<td>N/A</td>
</tr>
<tr>
<td>Primary fire:</td>
<td>Hit</td>
</tr>
<tr>
<td>Secondary fire:</td>
<td>Wind Up Hit</td>
</tr>
<tr>
<td>Rate of Fire:</td>
<td>1 hit/second</td>
</tr>
<tr>
<td>Reload Time (primary):</td>
<td>N/A</td>
</tr>
<tr>
<td>Reload Time (secondary):</td>
<td>N/A</td>
</tr>
<tr>
<td>Primary Damage:</td>
<td>20</td>
</tr>
<tr>
<td>Secondary Damage:</td>
<td>35</td>
</tr>
<tr>
<td>Notes: None</td>
<td></td>
</tr>
</tbody>
</table>

**VII. Levels**

**Sutter’s Mill: Above Ground**

Sutter’s Mill is one of the largest post-Collapse independent settlements in the former United States. Built among the wreckage of fallen California cities, the look of Sutter’s Mill mixes new and old technology, wreckage and renovation.

Sutter’s Mill’s above-ground appearance is spattered with heavy surveillance and defenses. Built into the landscape are terminals that outsiders can use to communicate with Lower Sutter’s Mill, the larger, subterranean portion of the settlement. Jekyll can use these terminals to communicate with Sutter’s Mill as he is no longer allowed into the lower levels.

The landscape surrounding Sutter’s Mill is rocky, mountainous, and dry. Vegetation does exist, although spread out and browned.

Sutter’s Mill is reminiscent of old west California but interlaced with new, homebrew technology. It is dirty, disheveled, but undeniably charming. Its defenses may be ominous, but there is a quality about the craftsmanship of the above-ground artifacts that suggests a unified, optimistic civilization fashioned them. It should be noted that the rugged
appearance of Upper Sutter’s Mill is primarily due to the fact that the reconstruction of the town has been heavily focused on the subterranean levels.

Sutter’s Mill is powered using old power generators established in the citadel-esque Reconstructed Church. While Jennifer and Jekyll are trying to establish a means of creating a more sustainable (not to mention defendable) power source for the settlement, a great deal of Sutter’s Mill’s power still comes from the generators. Sutter’s Mill is located close to the now dry American River, but receives water by pumping from the water table beneath the dry riverbed.

The design of Sutter’s Mill should contain wide open streets, alleys, collapsed buildings, rubble, burned out vehicles... anything that suggests that this was the site of an established city that was once destroyed. These items should be mixed with Steampunk-inspired surveillance equipment and defense weaponry. Standing buildings are acceptable as well, and a large, well-preserved church should be among them. From the back of the church should be a graveyard, home to a mixture of pre- and post-Collapse burials.

Ideas for nice creative touches for Sutter’s Mill may include: frayed propaganda/advertisement posters and unique debris (perfect for pop culture references).

The palette for Sutter’s Mill should be bright and pastel: oranges, yellows, pinks, purples, and browns. The level should feel sun-bathed, grainy, and oversaturated.

Upper Sutter’s Mill is the first level in High Noon, as it is where Jekyll is returned after infected with the Hyde parasite. Upper Sutter’s Mill will also be the scene for the final showdown with Boz Novak.

*Sutter’s Mill is a mixture of oranges, yellows, purples, and browns. Illustration of Sutter’s Mill by Lindsey Anderson.*
Reconstructed Church

The Reconstructed Church is the second level in the High Noon progression. The Upper Sutter's Mill level leads directly to the Reconstructed Church: a large, towering building among the wreckage of the settlement. Due to the fact that this structure is reconstructed, it consists of a frame, or bodywork, of a traditional church (denomination unimportant, put it is preferred to have some sort of gothic flavor), but be accented with definitively Steampunk embellishments on both the exterior and interior elements.

The Reconstructed Church houses the main power for a large part of Sutter's Mill. Though engineers have been building new generators within the protection of the subterranean Sutter's Mill levels, the recent attacks have halted production of these generators.

As the Reconstructed Church is itself a complete level, it should appear large and significant when viewed from the streets of Upper Sutter's Mill. It is multi-storied, containing a basement and two or three additional stories. On the interior, the Reconstructed Church should feel claustrophobic and dense in most areas, as well as broken and old.

Having one or two wide-open rooms (like a main hall or other central room) would certainly be a desirable element for larger (and more crowded) combat opportunities. A nice embellishment for a large, dramatic hall might be a pipe organ (with Steampunk flair, of course).

The lower levels of the Reconstructed Church should be blocked off by large, unbreakable doors (i.e. Hyde cannot break them) when the player enters the level. These doors should be opened from a lever on the third (or higher) floor.

Generators, functioning and non-functioning, should be located in large generator rooms in the basement levels, inaccessible until the player unlocks the aforementioned doors. One generator room should be obviously under assault, a bashed in door, perhaps, and a great deal of the generators in wreckage. The other should remain intact as the player attempts to fight off the onslaught of enemies in the Church.

The sepias, browns, and purples of the Upper Sutter's Mill color palette should also be present Reconstructed Church level. However, the significant decrease in light and introduction of stone and other historical building elements should alter the color palette to include grays, blacks, greens, and other cooler colors.

Lighting in the Reconstructed Church should be a mixture of light and dark, as certain areas will allow more light into the building than others via windows. Consider the effect of elements like stained glass on the color of your light and how effects the mood of the level. We have some great stained glass concepts that would look lovely in the church level, so light and shadow should certainly be forefront when implementing this level. As
the Reconstructed Church is on a failing power grid, it might be interesting to add flickering lights and even overloading circuits.

Humor is certainly invited when designing the Reconstructed Church level. Consider ironic elements like the idea of “worshipping technology.”

Pre-Collapse Graveyard (St. Mary’s Cemetery)

Stemming from the Reconstructed Church should be the Pre-Collapse Graveyard, St. Mary’s Cemetery, the third level in High Noon.

Like most of Sutter’s Mill, the Pre-Collapse Graveyard has been retrofitted with surveillance equipment and various defenses, which are now on the fritz due to the fact that the power generators have been damaged! This level should introduce an element of duck and cover to the player as he finds himself being attacked by his own security system! Strategically placed tombstones are a must for this level. The continuing sporadic power generators cause the Church’s back door to close behind the player, preventing retreat back into the Church. How convenient.

As this portion of the game takes place in the dusk/evening hours, the color palette for this level should be rich blues, greens, purples, and grays. This level is darker, so high-contrast lighting would be most appropriate. Like in the illustration to the left, clouds, a visible moon, and fog (emitters!) may be excellent additions for ambiance and mood in this level.

St. Mary’s Cemetery is remarkably old, with a history dating well before the Great Collapse. As such, the physical condition of the cemetery should suggest a certain level of disrepair. If possible, the addition of overgrown foliage would help to communicate this idea.

Like most things in Sutter’s Mill, St. Mary’s Cemetery has been partially reconstructed since the Collapse, and residents of the settlement have been using it to bury their dead for a great number of years. The design of the cemetery should reflect this fact. Research Steampunk ideas in regards to memorials, tombs, gravestones, etcetera, and apply them to the environment as you see appropriate. Most importantly, use your imagination! Due to the fact that it has been used as a cemetery for so long, St. Mary’s should be cluttered, and even disorganized in the placement of the tombstones and monuments.

There should be two to three accessible terminals so that Jekyll can readily communicate with Sutter’s Mill.

Keep in mind that Hyde can jump fairly high, so make sure the walls surrounding the cemetery are high and solid. We do not want the player feeling like they should be able to escape the cemetery by turning into Hyde.

High Noon Design Document v. FINAL
© 2007/2008 Chico State Game Studios
The Abandoned Train Depot

The Abandoned Train Depot is located beyond the boundaries of the Graveyard near one of the mountainous borders of Sutter's Mill, near an abandoned rail depot from the Pre-Collapse era.

The Abandoned Train Depot is discovered on accident by Jekyll, who is forced to find an alternate means of escape from the Graveyard after his security systems go haywire due to the events that passed in the Reconstructed Church level. A lucky find in an old crypt in the Graveyard reveals an old, once hidden tunnel that leads to a Pre-Collapse Rail Depot.

Novak has been using this Pre-Collapse Rail Depot as a place to station his robotic army and produce new units. As such, there are haphazard Mechanical Widow and Caterpillar assembly facilities that have been set up among the abandoned rail cars.

A Derelict Train is lodged in another tunnel in an adjacent mountain. The player should traverse the train through the cars to the head of the train to obtain the Spike-Eaze Rail Spike Gun.

The Train Depot is accessed through the Graveyard.
VIII. Flowchart

IX. Game Characters

Character Design

High Noon undeniably has an exaggerated feel to it, which seems appropriate given the nature of Steampunk adventure.

The characters are therefore simplified and stylized. It’s perfectly alright to mess around with proportions, exaggerated facial features, funny or silly costumes, comically large weapons, puns, pop-culture references, or anything else that strikes your fancy as a concept artist. Remember, the point of this project is not to depict realism, but to communicate story and style in a functioning (not to mention fun) game.

Like the palette for the levels, the color palette for the characters in High Noon should be bright, saturated, and stylized. That is to say, the muted, heavily textured look that a realistic piece might feature is not a desired look for this project.

Remember with both the modeling and the texturing: keep it simple. Think of your models like action figures (and no, I don’t mean Todd MacFarlane action figures!). We’re looking for that level of detail.

The Player-Character (Jekyll/Hyde)

Like the illustration below, the contrast between Jekyll and Hyde is dramatic. Jekyll is small, sophisticated, and even a little on the delicate side, while Hyde is brutish, large, and rugged. Jekyll is thin, with sharp-edged features (see illustration).

Hyde has an additional advantage over Jekyll in that he can jump significantly higher.
The Primary Antagonist: Boz Novak

Boz Novak is an opportunistic land baron who is intent on adding Sutter’s Mill to his growing empire. A model villain, Novak is dark-hearted, cut-throat, and only interested in one thing: money.

Boz Novak has one mechanical, steam powered arm that is wired to call in Steam and Armored Dragonflies, capable of bringing in all sorts of foes and obstacles for the player. Novak’s arm is also outfitted with a small, high powered gun in the event that his personal space feels compromised.

Monsters and Enemies

The monsters and enemies presented in High Noon are very flexible. Steampunk is fun like that: we can have mechanical enemies of all shapes and sizes, steam-powered enemies, and even some of the Hyde parasite soldiers. It all depends on how much we can get done in our tight timeframe. Below is the current lineup of High Noon enemies.

Mechanical Spiders

A nice, Headcrab-style enemy in terms of difficulty and encounter-frequency, the Mechanical Spiders make a nice, filler thing to shoot at, while not posing too much of a threat unless in great numbers (like those little Flood dudes in Halo).

<table>
<thead>
<tr>
<th>Mechanical Spiders</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Points</td>
</tr>
<tr>
<td>Damage</td>
</tr>
<tr>
<td>Range</td>
</tr>
<tr>
<td>Rate of Fire</td>
</tr>
<tr>
<td>Land/Air</td>
</tr>
<tr>
<td>Movement Speed</td>
</tr>
</tbody>
</table>

Because of their Headcrab/Flood-like nature, these Mechanical Spiders are commonly found in groups of three or more. Naturally, the more Spiders, the greater the threat, especially if other, larger, enemy types are present.

These spiders move very much like real spiders, and tend to scurry quickly towards the player when in attack mode. They attack using razor-sharp legs.

The Mechanical Spiders are smaller enemies; around three feet tall, receiving a significant upgrade from their UT2004 counterparts.
Mechanical Widows

![Image of Mechanical Widow]

An upgrade of the Mechanical Spiders, these deadlier spiders are larger than their weaker counterparts.

These spiders should exhibit more aggressive behavior than the Mechanical Spiders, in addition to being faster, and generally more menacing.

The Mechanical Widows are not what one would call powerful enemies, but due to their size they tend not to remain in clusters like the Spiders. These enemies can be considered lower-to-middle range in strength and threat level.

The Mechanical Widows use a small, mounted laser to attack from long range, in addition to possessing the razor sharp legs of their Spider kin. They are larger than the Mechanical Spiders, but not by much. They are around 3 feet in length.

<table>
<thead>
<tr>
<th>Mechanical Widows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Points</td>
</tr>
<tr>
<td>Damage</td>
</tr>
<tr>
<td>Range</td>
</tr>
<tr>
<td>Rate of Fire</td>
</tr>
<tr>
<td>Land/Air</td>
</tr>
<tr>
<td>Movement Speed</td>
</tr>
</tbody>
</table>

Steam-Powered Dragonflies

These flying enemies are certainly dragonfly-inspired, but are not completely faithful to the form of a real dragonfly. The important aspects to note about the Steam-Powered Dragonflies are that they are 1) mechanical, 2) steam-powered, and 3) between five to six feet in length.

![Image of Steam-Powered Dragonfly]

The Steam-Powered Dragonflies attack by shooting mounted turrets on both wings, which are structured in a biplane-like fashion.

The attack power of the Steam-Powered Dragonflies is not necessarily high, what makes these enemies challenging is the fact that they are hard to hit due to being airborne.

<table>
<thead>
<tr>
<th>Steam-Powered Dragonflies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hit Points</td>
</tr>
<tr>
<td>Damage</td>
</tr>
<tr>
<td>Range</td>
</tr>
<tr>
<td>Rate of Fire</td>
</tr>
<tr>
<td>Land/Air</td>
</tr>
<tr>
<td>Movement Speed</td>
</tr>
</tbody>
</table>

The dragonflies are pretty darn cool in that they can be used as flying scenery (non-hostile) in the sky in the open levels.

High Noon Design Document v. FINAL
© 2007/2008 Chico State Game Studios
Armored Dragonflies

Like the Spider/Widows, the Dragonfly has an upgraded form: the Armored Dragonflies.

The Armored Dragonflies have higher defense than the normal Dragonflies, and shoot more powerful ammunition. Their AI and animations should be similar to those of the normal Dragonflies.

In terms of difficulty for the player, the Armored Dragonflies are likely to fall under the "medium" category.

These are larger enemies, and should be between five to six feet in length.

Caterpillar

The Caterpillar should be a formidable foe: a train of armored units that shoots ordnance (grenades) toward the player.

This is a larger, stronger enemy type that is as rare as it is difficult to kill. Hyde is highly recommended to topple this baddie.

The Caterpillar is slow moving, not very aggressive, and generally a unit that sits in the back row and lobbs explosives at the player. These enemy types should be few to a level due to their level of difficulty and firepower.

Cars should explode and harm the player if he is too close when the enemy is defeated.

The Caterpillar is a large enemy; each car around 4 feet in diameter.
Lightning Bugs

The Lightning Bugs are formidable enemies. Powered by Tesla coils, they shoot bolts of lightning at the player for considerable damage. Lightning Bugs can fly for short bursts of time, but must land after a being airborne for around 4-6 seconds.

Unlike the Dragonflies, the Lightning Bugs are built with moving wings.

<table>
<thead>
<tr>
<th>Lightning Bugs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hit Points</strong></td>
</tr>
<tr>
<td><strong>Damage</strong></td>
</tr>
<tr>
<td><strong>Range</strong></td>
</tr>
<tr>
<td><strong>Rate of Fire</strong></td>
</tr>
<tr>
<td><strong>seconds</strong></td>
</tr>
<tr>
<td><strong>Land/Air</strong></td>
</tr>
<tr>
<td><strong>Movement Speed</strong></td>
</tr>
</tbody>
</table>

The Lightning Bugs are generally difficult to kill, possessing above average attack ability with reasonable defenses.

These are medium sized enemies, around 4 feet in length.

Grasshopper

Second in power to very few enemies, the Grasshopper is not a foe to underestimate. Swift, powerful, and agile, the Grasshopper is usually found alone or in pairs. Able to jump as high as Hyde, the aggressive Grasshopper can pursue the player practically anywhere.

<table>
<thead>
<tr>
<th>Grasshopper</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Hit Points</strong></td>
</tr>
<tr>
<td><strong>Damage</strong></td>
</tr>
<tr>
<td><strong>Range</strong></td>
</tr>
<tr>
<td><strong>Rate of Fire</strong></td>
</tr>
<tr>
<td><strong>seconds</strong></td>
</tr>
<tr>
<td><strong>Land/Air</strong></td>
</tr>
<tr>
<td><strong>Movement Speed</strong></td>
</tr>
</tbody>
</table>

The Grasshopper accurately attacks using a shoulder mounted turret that possesses considerable firepower, and fires at a significant rate.

Found in later levels, the Grasshopper is darker than most of the enemies in the High Noon bestiary, with glowing eyes and a generally creepy look to it.

A larger enemy, the Grasshopper is around five feet in height.
Friends and Allies

High Noon is designed in such a way that the player never actually sees any of the members of the Sutter’s Mill settlement. However, through dialogue boxes and/or voice acting, Jekyll will be able to interact with them through the security terminals installed in the levels.

The player primarily communicates with three Sutter’s Mill citizens: Jennifer Watts, an engineer who helped Jekyll design the security systems around Sutter’s Mill, Dr. Stephen Hudson, a scientist who specializes in parasitology and who is studying the effects of the Hyde Parasite, and Annie Gordon, a spy for Sutter’s Mill who works in Boz Novak’s research laboratories. See the Character Biographies section for more detailed descriptions of the NPCs in High Noon.

X. Story

Note: This section will have overlap form the Overview Section (particularly the “Backstory” section).

Complete Story (Summarized)

After Jekyll, the infected Sutter’s Mill citizen, is returned to the settlement, he discovers that he is no longer allowed back into the subterranean levels due to the fact that he will risk exposing the entire town to the parasite. Instead, Sutter’s Mill has decided to use Boz Novak’s parasite against him, and sends Jekyll in to head Sutter’s Mill first counter-attack.

Jekyll proceeds to fight off waves of Novak’s army through the streets of Upper Sutter’s Mill and through many of the Sutter’s Mill monuments: the Reconstructed Church, St. Mary’s Cemetery, and the Derelict Train. During these levels, Jekyll eventually learns that Novak has been inoculating himself against the parasite to ensure that he maintains complete control over his augmented army without risk of contracting the parasite himself. The coveted inoculations must be administered daily, meaning that Novak likely carries them with him at all times to ensure that he receives a dose when he should (and that no one else does).

Finally reaching the conclusion that if you want something done right, you must do it yourself, Boz Novak eventually agrees to face Jekyll at high noon in the streets of Sutter’s Mill to decide the fate of the settlement.

A formidable fight ensues, after which Jekyll has two choices: Join Novak as a member of his Hyde Parasite Army, or kill Novak and steal the inoculations (so that Sutter’s Mill can reproduce them), effectively neutralizing the Hyde parasite so he may return home.

Back Story

High Noon Design Document v. FINAL
© 2007/2008 Chico State Game Studios
After the Great Collapse, the United States crumbled into a land of warring city-states, renegade law enforcement, and ruthless, localized governments. Cities deteriorated as residents fled for safe-havens and settlements that promised protection. Some settlements became hideouts for scientists, academics, families, and others not suited for vigilante life in the wastelands. These settlements were usually well-hidden and heavily guarded; many stretching deep underground for maximum protection, and often hidden within the wreckage of former large cities. Most noteworthy of these settlements is Sutter’s Mill, a sprawling refuge in the former Sacramento Valley.

Settlements like Sutter’s Mill have become heavily sought after by a growing population of land barons, who, in the absence of a unified government, are ruthlessly usurping land to turn a profit from those seeking some semblance of civilization. The existing settlements are razed, scientists, technologists, engineers, and those with skills deemed of worth are captured and recruited, strengthening the land baron forces and making them a formidable power.

At the forefront of the land barons seeking to raze Sutter’s Mill is Boz Novak, a powerful baron who has managed to collect a large portion of the land in the mid-west. Seeking to stretch his influence to the former California coast, Novak has spent the last three years sending wave after fruitless wave of attacks on Sutter’s Mill’s formidable exterior defenses. Novak’s continued failures have caused him to turn to more drastic measures: parasitic warfare.

Using a recently discovered human parasite that causes severe episodes of rage and increased strength, Novak “enhanced” a percentage of his army to form a brute-force branch. As an added precaution, Novak managed to obtain and infect a member of the Sutter’s Mill settlement in an effort to destroy the colony from the inside.

**Narrative Devices**

The many story driven elements are to be delivered in two primary methods: through dialogue in-game between Jekyll and the citizens of Sutter’s Mill, and from cinematics.

We will not be making crazy, CG sequences for our cinematics. Rather, we will be using a trick that Neverwinter Nights employs to tell their stories: slow pans across stationary images in letterbox format (small, animated elements like hair or smoke are totally cool) with a narrator telling the story over them, and subtitles in the lower region of the letterbox.
XI. Character Biographies

Boz Novak

A rich land baron with an occasional flair for the dramatic, Boz Novak has been steadily usurping the settled and unsettled portions of the former United States for the past ten years. Though Novak initially started in the Midwest, he has steadily moved his attention to the former state of California in hopes of gaining control of the entire West Coast.

Boz both loves and fears the potential of the Hyde parasite. Regardless, he is unwilling to risk infection for fear of losing control of his formidable and growing empire. Novak’s workers have developed an inoculation for the parasite, to which Novak has exclusive access. The inoculations must be administered once daily, so Boz carries them with him at all times to ensure that he receives his dose at the correct time (and to ensure that only he has access to it).

Boz oozes confidence and control. He is always well dressed, cool, collected, and is basically the epitome of sinister villain. Boz’s motivations are purely monetary, and he rarely cares who he steps on to achieve his goals.

Jekyll (Player Character)

Jekyll has been an integral member of Sutter’s Mill for most of his life, and is close friends with many of its inhabitants. He is tall, thin, and a bit awkward, but generally liked and well-accepted among those at Sutter’s Mill.

Since Boz Novak starting attacking the Sutter’s Mill settlement, Jekyll and Jennifer Watts have been working non-stop on designing new armaments and surveillance equipment for Upper Sutter’s Mill. Jekyll’s unfortunate run-in with Boz Novak that led to his infection with the Hyde parasite happened during a day when he was above-ground upgrading surveillance equipment and terminals.
Jekyll was not targeted directly, but was in the wrong place at the wrong time. News of his infection hit shortly after it happened when Annie contacted Sutter’s Mill with knowledge that Novak had plans to infect a member of the settlement.

**Annie Gordon**

Annie Gordon is not from Sutter’s Mill, but rather from one of the many settlements usurped by Boz Novak’s steadily growing forces. Annie agreed to join forces with Novak and works in his research laboratories that are responsible for the discovery of the Hyde parasite, but only did so out of fear for what Novak would do if she declined. Annie maintains a sour hatred for Novak, and frequently supplies information to targeted settlements in hopes of preventing what happened to her hometown.

As such, Annie has been working as a spy for Sutter’s Mill, and spends a great deal of her free time passing information to Dr. Hudson in hopes of helping the settlement defeat Boz Novak’s forces, or at the very least deterring them.

Annie is sharp-witted, intelligent, and extremely perceptive. She can be quick to judge, and has a temper that runs notoriously hot. Despite these facts, many find Annie incredibly charming.

Annie is tall, in her late 20’s, and has blonde, shoulder-length hair. She has piercing blue eyes and pale, freckled skin.

**Dr. Stephen Hudson**

Dr. Stephen Hudson is a round, balding man with a gruff, no-nonsense demeanor. A scientist by trade, Dr. Hudson’s forte lies in parasitology, and proves to be an invaluable asset to Jekyll post-infection. Dr. Hudson works closely with Annie Gordon through secret, encrypted communications.

Having recently moved from another settlement to Sutter’s Mill, Dr. Hudson has few friends and spends a great deal of time researching the effects of Hyde parasite.

Visually, I see Dr. Hudson as a cross between Zidler from Moulin Rouge (Google it if you don’t know what I’m talking about) and the Overseer for Vault 13 from Fallout 1 (again, Google is your friend). In personality, I that mixing equal parts Zidler, the Overseer, and Grissom (CSI: Las Vegas) would be an adequate representation of Dr. Hudson’s charm (or lack thereof).
Jennifer Watts

Jennifer Watts moved to Sutter’s Mill with her family as a child shortly after the end of the Great Collapse. An engineer and general tinkerer, Jennifer worked with Jekyll to set up the newer elements in the Upper Sutter’s Mill defense system.

Jennifer’s father died many years ago due to illness, leaving her with her mother for many years. However, Boz Novak’s recent attacks have cost Sutter’s Mill a number of lives, with Jennifer’s mother among them.

Jennifer is dark skinned and has hazel eyes. She has curly black hair, and is of average height. She is good natured, and has much of her father’s natural curiosity for exploring how things in the world work. She is likely in her mid-50’s.
XII. Custom HUD

Proposed Elements

1. Directional hit indicators to communicate to the player from which direction they are being attacked.
2. A “message zone” for messages such as “Press “E” to activate” or “Picked up a Red Shirt’s Revenge Ray Gun.”
3. A numerical health system and portraits indicating whether Jekyll or Hyde is the active character.
4. A reticle (duh).
   A blocked off area that represents the current ammo and reserve ammo in terms of numbers.
5. Two vials that are grayed out when the player does not have any Serum Vials, and turn green when the vials are picked up. Two vial maximum.
Welcome to the High Noon team! Prepare to work hard, learn a ton, and meet some amazing people! Oh, and make a video game! Woot!

Overview
Activity: Tuesday/Thursday
6:00 – 7:50 PM
OCNL 251
Lecture: Friday
2:00 – 2:50 PM
OCNL 254

Alisha Thayer
Director, High Noon
Creative Consultant, Davis
alishathayer@gmail.com

Andrew Holsfield
Director, Davis
Technical Advisor, High Noon

This class has been designed to emulate a production studio in the video game industry as closely as possible in an academic environment. That being said, this class is very difficult and time consuming. Please carefully evaluate your schedule during the first few weeks of school. If you are taking a heavy production load, please talk to me as soon as possible.

Attendance
Attendance in this class is MANDATORY. As we are all working in a team, it is imperative that all team members are present during production. Absent team members lead to missed deadlines and miscommunications. As this is a senior-level production class, "free" absences are not allowed. However, if you must miss a class, please e-mail the Director before class so we can discuss your absence.

Grading
Your grade in this class will be broken up into the following segments:

Attendance: 25%
Self-Evaluations (Mid-term and Final): 50%
Director Evaluation: 25%

Self-evaluations are extremely important to me as it is very difficult to grade this class. You are expected to be honest and forthcoming on your self-evaluation.

Google Group/Class Communications
Communications with your fellow team members is strongly encouraged outside of class. Please use all methods of communication that you have available to you. Check your e-mail often. If you have any IM information, please exchange it with people you work closely with in the class. You will be amazed how easy it is to get your work done when you can get those annoying little questions answered on the spot!

Please note I have given you my phone number, e-mail address, and IM information for MSN, Yahoo, ICQ, and AIM. You are strongly encouraged to use these to contact me at any time regarding any questions or concerns you have about High Noon. I keep my phone on my nightstand at night; if it's 3:00 AM and you have an urgent question, you have an open invitation to call me.

You are required to join the class Google Group, which is located at: http://groups.google.com/group/2008-2009_CSGS_HN

Through this group, you will have access to contact information for the entire team (phone numbers, IM information, e-mail addresses), as well as class materials such as our design document, class handbook, etcetera.
EXAMPLE ASSESSMENT EXERCISES

Davis Concept Design Assignment 1

Part 1 of 2: Research

Alisha Thayer
Art Director, D.A.V.I.S
alishathayer@gmail.com

Welcome to D.A.V.I.S! As a concept artist, you will be shaping the visual style of the game and providing a visual blueprint for modelers and level designers as they produce and implement assets. Your contribution to D.A.V.I.S. is vital. Even the most minor of design decisions can and will permeate through the images and content generated by the game. Think about your design decisions carefully, but have fun with it too! Want to see your name on the bowling alley downtown? Put it there! Make this game your own, make it beautiful, and make it relevant to the established visual style.

Before you can make art for D.A.V.I.S., it is important that you research the time period in which the D.A.V.I.S. world exists. Use the following two prompts for Act 1 and Act 2, in combination with the existing art on the bulletin board and the knowledge that this game takes place in a dilapidated 1950's facility made to look like a perfect exterior city while you conduct your research.

Act 1: Suburbia

The first act takes place in the gated community sitting on a hill overlooking the rest of town. It was constructed in 1952 to be the utopian American small town of the 1950s. This is the area of D.A.V.I.S. where the families of the spy training program lived. When the simulation was terminated, all the doors and windows of every house in Suburbia were simultaneously locked with air-tight metal doors and the houses were filled with poisonous gas, killing all the spy trainees. The facility was then abandoned for years and years until Katherine came back and started to reshape it. Being only one person trying to maintain an entire town, most of her changes are sloppy and are limited to those areas most important to her.

Suburbia is dominated by faded pastel colors, brown & dying grasses, orange/yellow tumbleweeds, and cracking grey streets & sidewalks. The houses are all identical or mirror images of each other. Their lawns are overgrown weeds instead of freshly cut grass, the shrubs are over grown, there’s ivy climbing up the sides of the houses, and moss clings to the siding. The sidewalks and streets are cracked and dusty. The only exception is Katherine’s house. Her lawn is more recently cut but it’s still brown, dry, wild grass. The paint on her house is fresh, but it was put on poorly, unevenly, and has even dripped in some places.

Suburbia is barren and quiet. There are no birds chirping, no cars, and no sounds of people whatsoever. After the crescendo from the title overlay, the music goes very quiet. Slowly the music will return, but it will stay very light and eerie. Occasionally the player will walk near an electrical box that hums softly.

Act 2: Downtown

The second act takes place in the downtown area of D.A.V.I.S. in the middle of the facility. The downtown area is where the spy trainees would work, shop and have “recreation time.” The center of the town has a medium sized rotary with a statue of the D.A.V.I.S. logo.

This area of D.A.V.I.S. is filled with buildings such as the diner, town hall, the post office, a general store, the gas station, etc. The buildings were originally painted with bright, saturated primary and secondary colors, but have since faded over the years. Downtown is bordered by four distinct obstacles: on the side of Suburbia is a cliff with Suburbia being above Downtown; on the side of Backstage is a river; the third side is another cliff, but this time leading down into the ocean; the final side is a hill with the train tracks running along it 40-50 feet up. The Diner is near the river and bridge leading to Backstage; the Drive-In is located near the hills; and the cliff is where the look-out point will be. There are cameras hidden in various places throughout downtown; in bushes, mannequins, signs. The mannequins in this area have also been personalized by Katherine. Music will be coming from the PA system, radios, stores, and TVs.

233
Part One: Getting to Know Your Team

Select your two favorite images from the images you submitted for this assignment. Place them on the X drive on the ECST Server (ask me how if you need help) in the following directory: X:/APCG 495/Concept Design/Assignment 1/Sample Images/YOUR_NAME. Be prepared to present these images to the rest of your team, explaining briefly why you consider your images to be significant. Please make sure to introduce yourself before you present your images.

Part Two: Selecting a Region

Log in to the D.A.V.I.S. GoogleGroup and find davismap.jpg. After I've explained the format of the map and notable landmarks in each of the regions, please select which region you'd be most interested in envisioning. We will be assigning one person per region, so be prepared to default to a backup region if your selected region is taken.

Part Three: Applying Research

Copy the master zip file of everyone's contributions from X:/APCG 495/Concept Design/Assignment 1/CD_AssignmentArt.zip to your computer. Take ten minutes to go through the images that your teammates selected and note the images that will help you design your region. Also note who submitted the images you've selected; you're encouraged to ask them questions about where they found their images or what search terms they used.

Part Four: Envisioning D.A.V.I.S.

Our first major task in D.A.V.I.S. will be to lay some groundwork down regarding the content of D.A.V.I.S. as a city. Look at your region carefully, as well the regions neighboring yours. Note who is working on those regions. These people are your design buddies. It is your responsibility as a designer to ensure that your decisions regarding architecture, building types, level of dilapidation, decoration, etc. are consistent with those of your neighboring regions. Please use your design buddies to bounce ideas off of as you explore and establish the characteristics of your region.

Assignment Details:

Your assignment is to flesh out all of the major visual details of your region. Questions you could ask yourself about your region include:

- What (if any) businesses are in my region? What are they called?
- What condition are the buildings in?
- Are there advertisements in my area? What do they say?
- What makes my region unsettling?
- What areas in my region could belong to Katherine? Why?
- What do they look like? How are they different from the other areas in my region?
- What (if any) kind of houses are in my area? What do the front yards look like? What about the post-boxed Doors?

You are to turn in the following items to X:/APCG 495/Concept Design/Assignment 1/YOUR_NAME:

- A Word Document containing an index of important landmarks, buildings, and other possible art assets in your region.
- Sketches of important landmarks—please ensure that your sketches are clean and professional JPEG format is sufficient for this assignment.
- A bank of images that illustrate the important visual aspects of your region, such as: road condition, building condition, building decorations, mannequin positioning.
- Anything else you feel illustrates the look and feel of your region.

This assignment is due THURSDAY, FEBRUARY 5th at the START OF CLASS.
Assessment Assignment:

If you haven’t already, please carefully read the back story for High Noon (Design Doc, pages 6-9) and the setting description for the Sutter’s Mill level (Design Doc, pages 15 – 17). Also make sure you have visited Deviant Art and Google Images for good Steampunk references.

Based on what you have read, you will be creating an animated walk cycle for the Mechanical Spider, Grasshopper, or Mechanical Widow.

- You will be supplied with a model of the Mechanical Spider, Grasshopper, or Widow, with which I expect you to create your walk cycle. I’m not looking for perfection, but I do want to see an understanding of timing and weight.
- The model will have an existing rig, so please do not worry about rigging at this time.
- Please ensure that no more than one person work on an enemy type. That is to say, if someone is working on the Grasshopper, please work on the Widow or Spider. Unless, of course, we have more animators than walking enemy types.
- You may work in the animation package that you are most comfortable with for the time being.

You are to have your animation complete and rendered by the beginning of class, Tuesday, September 8th.

You will be turning in the following items:
- The object/animation file
- A movie render of the walk cycle

Your team **Technical Director** is: Winston Thrasher

Your team **Research Lead** is: Scott Washington

Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual, though your work does.
Assessment Assignment:

Character Concept Artists/Game Manual & Promos

If you haven’t already, please read and understand the following (you will find overlap in some sections):

Design Document:
- Overview/Background (Pages 6-9)
- Levels (Pages 15-20)
  - Game Characters (21-22)
  - Monsters and Enemies (Pages 22-26)
- Story (Pages 27 – 28)
- Character Biographies (Pages 28-30)

The Art of High Noon
- Characters (Pages 8-17)
- Enemies (Pages 30 – 29)

High Noon Concept Art Inspiration Picture Book

You can find all relevant documents on the GoogleGroup.

Based on what you have read, please illustrate one of the following:

1) An attack on Sutter’s Mill.
2) Propaganda for New El Dorado, Novak’s growing empire.
3) A metropolis of New El Dorado (one of the fallen and rebuilt cities under Novak’s control).
4) Jekyll awakening in Sutter’s Mill after being infected with the Hyde Parasite.
5) An advertisement for one of the weapons in the High Noon arsenal.
6) A Steampunk-inspired advertisement for something whimsical of your choice.

You have complete creative control on interpretation of the characters, scenery, color palette, and dramatic angles (if any), but the image should be truthful and accurate to the existing story. Characters/enemies should be recognizable in your image, so please use existing character designs when creating your image. You may include as many or as few characters/enemies in your image depending on your comfort level with character drawing versus environmental drawing.

You can also find artwork related to High Noon at the following sites:

This image is expected to be professional quality, color, and at no less than 1000 pixels in width. Please illustrate all images at 300 dpi.

You are to have this image complete by the beginning of class, Thursday, September 11.

You will be turning in the following items:

- Any sketches you have done
- Your completed image in PSD and JPG formats

NOTE: Please refrain from flattening images—I want to see your process in your Photoshop file.

Texturing

If you wish to instead assist Modeling/Texturing this semester with texturing, please meet with me after class today. We will set up some time outside of class to talk about techniques and brushes used in this class to achieve the look of the game. You will then be given a model and a UV snapshot, and be given one week to complete a refined, appropriate texture for your model.

Your team Technical Director is:
Eric Keenan
Technical Director
AIM:...

Your team Art Director is:
Alisha Thayer
Art Director
ailshathayer@gmail.com
AIM:...

Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual, though your work certainly does. I encourage and require teamwork in this class.
Level Designers

Assessment Assignment:

Level Designers, welcome to Level Design Boot Camp! You are to spend your first week with High Noon completing a series of introductory UT3 videos that are located here:

http://www.utforge.com/index.php?PHPSESSID=ccltl11u8eomitlo6p86q0bklvsm0&page=31

These videos are designed to familiarize you with the Unreal Editor so that you feel comfortable working with the tools necessary to designing a good, functioning level with U3 Technology.

For this assignment, please view and follow videos 1-10 in the *Official UnrealEd Training Videos: Chapter 1 - Getting Started* section.

You will be completing tutorial levels to start. You will be saving your level as a .ut3 file and turning it into me at the end of this assignment.

You are to submit your tutorial files by the beginning class on Thursday, September 4th.

PLEASE NOTE: These videos have a few broken links that can be navigated around by doing the following:

- Videos 3 & 4, titled “Basic Level (Part 1)” and “Basic Level (Part 2)” do not work
- To access them, open “Basic Level (Part 3)” and click play to start buffering it. Press pause so that the video keeps buffering, but does not play.
- When the video is fully buffered, slide the time indicator to the end of the video. This will bring up 3 links to related videos.
- “Basic Level (Part 1)” should be on the right and “Basic Level (Part 2)” should be on the left

Your team **Technical Director** is:

Bryson Perez

Your team **Research Lead** is:

Tom Ruiz

Aim:

*Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual, though your work certainly does.* I encourage and require teamwork in this class.
Assessment Assignment, part 2:

Level Designers, welcome our second chapter in Level Design Boot Camp!

Your next task will be viewing in-depth material on level construction. This is practical experience with the level editor that will have you building a much more sophisticated level from scratch. This is an involved tutorial that will take around 6+ hours to complete, and contains many important concepts that we will be using when we implement the High Noon levels.

You will get this set of videos from me in class, as they are not on the website.

Your completed level from these tutorials will be due at the beginning of class on Tuesday, September 16th.

In addition, you are to spend the next week following the “Official UnrealEd Training Videos: Chapter 2 - Introduction to UnrealEd” videos located here:


You are to view and follow videos 1 – 35. Please note that these tutorials, while short, add up to about 3.5-4 0 hours of content. Please ensure that you allow enough time to view and understand the content presented in the videos. For most of you, they should be familiar concepts, and really shouldn’t take the full 4 hours to learn.

You will have a skills check on the topics presented in the UnrealEd Training Videos, Chapter 2 on Thursday, September 11th.

PLEASE NOTE: Part of your lab time on Tuesday, September 9th will be spent in lecture, as you will be covering level design best practices with Tom Ruiz. Please consider this when planning your approach to watching the video tutorials.

And, as a friendly reminder . . .

Your team Technical Director is: Bryson Perez

Your team Research Lead is: Tom Ruiz

Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual, though your work certainly does. I encourage and require teamwork in this class.
Assessment Assignment:
If you haven’t already, please carefully read the back story for High Noon (Design Doc, pages 6–9) and the setting description for the Sutter’s Mill level (Design Doc, pages 15 – 17). Also make sure you have visited Deviant Art and Google Images for good Steampunk references. Based on what you have read, you will be modeling one of the following building types:

- Café
- Saloon
- Train Station Platform
- Barn
- Abandoned House
- Water Tower
- Shed
- Bank
- Factory
- Rubble: this building has been destroyed beyond recognition and did not have an opportunity to be rebuilt before Sutter’s Mill went subterranean.

You may use the modeling package you are most comfortable with for this assignment.

You are to design, model and texture this building by the beginning of class, Thursday, September 11th.

Model requirements (Your model MUST follow each of the following):

Geometry:
- That it is tripled.
- That the model does not contain any inverted normals.
- That the model is of an appropriate polygon count (1000 triangles for this assignment)
- That there exist no rogue n-gons or quads.
- That the geometry possesses an appropriate flow of polygons, and that there are no parts of the model that have too few vertices, to ensure the model will light properly.
- That there are no intersecting polygons.
- That the file name contains no spaces and is named logically. (SM_Building_PROPNAME)

UV Maps and Textures
- That its UV contains no distortion by applying a checkerboard map. (Google)
- That there is only one UV map per model. Likewise, there is only one texture per model.
- That the texture for the model is 2048 x 2048 in size.
- That no visible seams exist in the texture when applied to the model.

Overall
- That the model is aesthetically pleasing and fitting the style of High Noon.
- That the texture is aesthetically pleasing and fitting to the style of High Noon.

You will be turning in the following items:

- Sketches of the building. Please ensure these are done before you start modeling.
- The object file in a format native to the package you used
- Your texture in PSD format (please preserve layers as I want to see your process)
- Your texture in JPEG format (for quick viewing)
- A screenshot of your UV map (in JPEG format)

Your team **Technical Director** is: Eric Keenan

**Research Co-Leads:** Andrew Holfried

Jeremy Maligrig

AIM: [Contact information]

Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual, though your work should be.
Assessment Assignment:

Your assignment is to select two already existing weapons in the UT3 arsenal. You are to switch the ammo types by exploring the code and modifying where necessary.

For example, you could have the Enforcer shoot Link Gun rounds.

You are to submit this modification, complete, by the beginning class on Tuesday, September 8th.

You will be turning in the following items:

- A demonstration of the switched ammo types in UT3
- A compressed archive of your code.
- An explanation of your process (1-2 paragraphs) in a word document. Please put this file in the compressed archive with your code.

Your team Technical Director is: Michael Sakuma

Your team Research Lead is: John Oilar

Please be sure to consult with them if you are in need of assistance. This assignment does not need to be individual. I encourage and require teamwork in this class. Working with your friends is NOT CONSIDERED CHEATING. Please use all resources you have available to you to get the job done.
APPENDIX F
LEVEL DESIGNER NAME HERE

- Navigate the orthographic views.
- Navigate the perspective view.
- Test the level by activating “play from here.”
- Tumble the camera around a local pivot point. Zoom on an object.
- Snap the camera to an object.
- Turn on and off real-time in your perspective view.
- What does “unlit movement” do?
- Explain “texture density mode.”
- Explain “shading density mode.”
- “Game Mode” refers to what? What is it mapped to on the keyboard?
- Demonstrate “Lock Selected Actors to Camera” with a spotlight.
- Explain the occlusion parent/child relationship in the viewports.
- Show me the “show flags” button.
- Explain the difference between wireframe and brush wireframe mode.
- What is “squint mode”? What would you use it for?
- What’s the difference between an additive map and a subtractive map?
- Explain what “Show Only Modified Properties” does from the view dropdown menu.
- How do you access the texture properties for a BSP? (F5)
- Show me how to change your grid size.
- Change your viewport configuration for me.
- What’s detail mode?
- What’s a prefab? What does it mean to lock prefabs?
- A group of actors— you can create prefabs for quick placing of complex, multi-actor objects.
- If you see a blue wireframe around a BSP, is it additive or subtractive? (additive)
- Go down each of the top four options in the “Brush” dropdown menu. Explain to me what each does.
- What does the import function do under the brush dropdown menu?
- Your Map Check is telling you that a brush has a NULL material reference. What does that mean?
- What’s a pickup light?
- Modify the far clipping plane in the editor.
- You remember that using the UNDO function in UnrealEd is really, really sketchy, right?
- Explain to me the benefits of preventing the mouse from being to do anything but select in the level. Now show me how to do it.
- Show me the non-uniform scaling widget. Now agree with me when I say it kicks butt.
- Explain the difference between local axes and world axes.
- Access the Generic Browser for me.
- Access Texture Alignment mode and use it for me.
- Turn the grid on and off.
- Change the camera speed in the editor.
- Bring up the options for a cube BSP brush.
- What do CSG actions effect?
- Show me the “Show All Actors” button.
- Where can you look in the editor to tell you how many static meshes you have selected at one time?
- What’s a DrawScale? How do you modify it?
- Say you want to turn off autosave (God forbid). How would you do that?
- Show me a quick way to modify your grid and rotation grid settings.
- Make a light and change its color.
- Change the texture on a BSP.
- Demonstrate the “Search for Actors” button.
- Find and place a static mesh.
- For the purposes of level design, what is an actor?
- Modify the shape of a builder brush with the geometry tools.
- What is soft selection mode? What is it used for? Can you UNDO a soft selection modification?
- Extrude a builder brush for me.
APPENDIX G
Stained Glass Windows
Ayla Richards
October 2007

A Model Sheet from High Noon Semester One by Concept Designer Ayla Richards.
A Model Sheet from D.A.V.I.S. Semester One by Concept Designer Lauren Kodai.
APPENDIX H
MODELING CHECKLIST

STEP ONE: Checking Geometry and UV Maps/Adding New Geometry
Ensure that the model you are given has:

- Tripled geometry.
- No inverted normals.
- No rogue n-gons or quads.
- No intersecting polygons.

Also ensure that the model’s UV map meets the following requirements:

- No distortion. Apply a checkerboard map to the model and check for distortion in the UV. Checkerboard maps can be easily found using Google Images.
- Proportional UV’s. Check that large items on the model are represented by equally large UV coordinates on the UV map.

After you have checked that the model is ready for production, you may add elements to the geometry to ensure it is an appropriate level of detail for the new engine. You are welcome to be creative and expressive during this process, but please stay within the polygon limits for your type of model.

- If you are given a building, you may build between 800 and 2000 triangles.
- If you are given a small prop, please do not exceed 500 triangles.
- If you are given a medium-sized prop (larger than ½ player size or smaller than a small building), please do not exceed 1000 triangles.
- If you are given a first-tier asset (and you’ll know if you have one), then you are expected to use between 2000 and 10000 triangles.

If you have added geometry to the mesh, please ensure that you add the new geometry to the existing UV map, checking for distortion, proportion, etc, before proceeding to texturing.

STEP TWO: Texturing
Almost all of our models now require a texturing upgrade to reflect the new capabilities of the UT3 engine.

- If you are given a building, your texture will be 2048 x 2048 pixels. Your model may also be multiple parts (as long as they’re large) containing more than one 2048 x 2048 texture.
- If you are given a small prop, your texture will be 1024 x 1024.
- If you are given a medium-sized prop (larger than ½ player size or smaller than a small building), your texture will be 2048 x 2048. Please do not exceed more than one UV map/texture.
- If you are given a first-tier asset (and you’ll know if you have one), you may use multiple 2048 x 2048 textures to achieve the highest quality effect.

You are to use techniques learned in class and internet resources to texture your model. Use the techniques presented in the Next-Gen Texturing tutorial I gave you in class, ignoring all the stuff about 3DS Max. Use fellow artists in class to learn new techniques. Be critical of your work and try to make it look as appropriate for the game as possible. Look at old artwork for High Noon for inspiration.

STEP THREE: Checking Scale and In-Game Textures
Your last task will be to check that the scale of your model translates accurately into Unreal 3. You will be given a separate tutorial on how to navigate Unreal Editor, including importing a static mesh into a temporary package, importing a texture, creating a new material, assigning a diffuse texture, and placing a static mesh in a level. If you find that you need to adjust the scale of your model, please do so to the model itself and then re-import into Unreal.
APCG 495: Advanced Production
Mid-Term Self Evaluation
DUE THURSDAY, MARCH 12 BY THE END OF CLASS

Name:________________________________________
Date:_______________________________________

1) Please rate your attendance in this class. (Circle ONE)
   Perfect (0 absences)  Excellent (1 absence)  Good (2-3 absences)  Needs Improvement (more than 3 absences)

2) Please rate your work ethic in this class. (Circle ONE)
   Perfect  Excellent  Good  Needs Improvement

3) Please rate your ability to meet deadlines in this class. (Circle ONE)
   Perfect  Excellent  Good  Needs Improvement

4) Please evaluate approximately how many hours you have allocated to this project outside of class per week. (Circle ONE)
   10+ hrs/week  8-10 hrs/week  6-8 hrs/week  4-6 hrs/week  less than 4 hrs/week

5) Please rate your workload in this class (Circle ONE)
   Extremely Heavy  Heavy  Moderate  Light  Very Light

6) Please evaluate your performance in this class. (One sentence minimum.)

7) Please indicate the letter grade you feel you deserve at this point in the class.
   A  A-  B+  B  B-  C+  C  C-  D+  D-  F

8) Feel free to use this section to let me know anything you have to say about your experience with D.A.V.I.S. thus far!
The Chico State Game Studios Team

High Noon: Semester One

Executive Producer
Clarke Steinback, Ph.D.

Writer, Director & Producer
Alisha Thayer

Assistant Director
Dan Schimmitou

Technical Director of Art
Adam Bava

Technical Director of Animation
Russell Henry

Technical Director of Programming
Michael Sakuma

Technical Director of Level Design
Tom Ruiz

Art Assets Team: Concept Design
Ramon Aguilar
Gary Angeja
Pablo Barriga
Tiffany Comfort
Ayla Richards

Art Assets Team: Modeling/Texturing
Adam Bava
Christopher Colton
Andrew Hollfield
Brian Howland
David Winston

Animation Team
Steven Beyers
Russell Henry
Winston Thrasher

Programming Team
Chris Mauzey
Stephen Munoz
John Oilar
Michael Sakuma
Bob Tellez

Level Design Team
Drew Barrows
Thomas MacFarlane
Tom Ruiz
Bob Tellez
Ian Williams

Music
Scott Jason
Dylan Smith
Jon Starrett

Cinematics
Lindsey Anderson
Valerie Rodriguez
Ayla Richards
Gary Angeja
Roxanne Barrett
Alisha Thayer

Web Design
Ashley Kremenlik
Alisha Thayer

Support Staff
Lindsey Anderson
Dan Krieger
Scott Washington

Inspired by "The Game With No Name" Game
Concept by
Evan Woolery
Executive Producer
Clarke Steinback, Ph.D.

Writer, Director & Producer
Alisha Thayer

Assistant Director
Dan Schmittou

Interim Director
Andrew Holifield

Technical Director of Art Assets
Andrew Holifield

Technical Director of Animation
Winston Thrasher

Technical Director of Programming
Michael Sakuma

Technical Director of Level Design
Tom Ruiz

Art Assets Team: Concept Design
Ramon Aguilar
Pablo Barriga
Gary Angeja
Tiffany Comfort
Jade Reed
Ayla Richards

Art Assets Team: Modeling and Texturing
Sonya Ballas
Tou Lee Chang
Christopher Colton
Ryan Crangle
Andrew Holifield
Eric Keenan
Ron Lehman
Jeremy Maligie
Sou Saecho
Stephen Wanders

Level Design Team
Drew Barrows
Thomas MacFarlane
Tom Ruiz
Ian Williams

Programming Team
Chris Funk
Chris Mauzey
Stephen Munoz
John Ollar
Michael Sakuma
Dan Schmittou

Animation Team
Steve Beyers
Eric Flores
Winston Thrasher
Alan Whyder

Animation Advisor
Scott Washington

Programming Advisor
Bob Tellez

Texturing Advisor
Lindsey Anderson

Graphic Designer
Mario Magliozi
**Executive Producer**  
Clarke Steinback, Ph.D.

**Writer, Director & Producer**  
Alisha Thayer

**Art Director**  
Alisha Thayer

**Technical Director of Art Assets**  
Eric Keenan

**Research Lead, Art Assets**  
Andrew Holifield  
Jeromy Maligie

**Technical Director of Animation**  
Winston Thrasher

**Research Lead, Animation**  
Scott Washington

**Technical Director of Programming**  
Michael Sakuma

**Research Lead, Programming**  
John Oiliar

**Technical Director of Level Design**  
Bryon Perez

**Research Lead, Level Design**  
Tom Ruiz

**Art Assets Team: Modeling**  
Tou Lee Chang  
Justin Ellis  
Ji Her  
Tyson Hill  
Aaron Kaplan  
Eric Keenan  
Raina King  
Jameson B Marrs  
Ryan Nichols  
Teng Xiong

**Art Assets Team: Concept Art**  
Gary Angeja  
Roxanne Barrett  
Ferdinand J. Cosico  
Joshua Gabriel  
Jade Reed  
Ayla Richards  
Mike Shanks  
Alisha Thayer  
Stephen Wanders

**Animation Team**  
Sonya Ballas  
David Brooks  
Eric Flores  
Winston Thrasher  
Scott Washington

**Level Design Team**  
Andrew Dutra  
Bryson Perez  
Cole Reiben  
Tom Ruiz  
Nathan Smart  
Ian Williams

**Programming Team**  
Michael Brennock  
Chris Funk  
Matt Hill  
Paul Hoerman  
Stephen Munoz  
John Oiliar  
David Saesere  
Michael Sakuma  
Robert Utter  
Paul Weatherby

**Audio Team**  
Catherine Arthur  
Matt Copeland  
Jonah Hontos

**Writing Team**  
Mike Metz  
William “Bill” Saporito

**Director, D.A.V.I.S.**  
Andrew Holifield

**Assistant Director, D.A.V.I.S.**  
Jeromy Maligie

**Creative Team, D.A.V.I.S.**  
Lindsey Anderson  
Nathan Harling  
Andrew Holifield  
Jeromy Maligie  
Alisha Thayer  
Robert Utter

**Texturing Advisor**  
Lindsey Anderson

**Graphic Designer**  
Amanda Kiefer  
Mario Magliozzi