5-2012

THE ADAPTATION OF 3D VIDEO GAME PRODUCTION TECHNIQUES TO THE 2D GAME PLAY MECHANICS OF THE BEAT-EM-UP GENRE

Donald Roeder
Clemson University, dproeder@gmail.com

THE ADAPTATION OF 3D VIDEO GAME PRODUCTION TECHNIQUES TO THE 2D GAME PLAY MECHANICS OF THE BEAT-EM-UP GENRE

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Masters of Fine Arts
Digital Production Arts

by
D. Patrick Roeder Jr.
May 2012

Accepted by:
Dr. Brian A. Malloy, Committee Chair
Dr. Timothy A. Davis
David Donar

Abstract

In this paper, we address some of the technological issues and shortcomings that led to the recession in popularity of the *Beat 'em up* video game genre. In particular, we adapt current technological advances and apply them to 3D video game development to mirror the mechanics of the 2D *Beat 'em up* genre. We describe our assembly-line approach to 3D asset production that permits quick development of multiple and varied characters and environments. Our approach to 3D camera placement provides the same viewpoint used in the 2D game approach. We have developed various scripts to streamline the animation process. In addition, levels are designed retroactively with a "go right" mentality, and various programming techniques that facilitate swappable, and multiple enemy types with different AI behaviors. Finally, we apply current hit detection technology to the spirit of the *Beat 'em up*.

Table of Contents

List of Figures

Figure Page

Page

List of Figures (Continued)

Page

Chapter 1

Introduction and Motivation

The *Beat 'em up* video game genre, also known as *the brawler*, is a genre that generally features hand-to-hand combat between one or more player characters against a large number of non-playable enemy characters (NPCs) [1]. These games typically occur in an urban setting and feature conflict between a team or loose group of friends against unified and overwhelming forces of evil. Traditional beat 'em games contain scrolling, two-dimensional (2D) stages, punctuated with pockets of enemies that must be defeated before the player can progress through the level. The most notable feature of this genre is the elementary chaining of attacks while managing multiple enemy threats from many directions. The simplistic gameplay of this genre makes them appear somewhat shallow in plot and gameplay experience, yet typical advocates of this genre become completely invested in the pick-up-and-play nature of this light and enjoyable gameplay experience. In today's climate of casual gameplay, this genre has never been more relevant and a worthy genre for propagation to 3D technology.

However, the past decade has begun with a sharp decline in public interest in the pure *Beat 'em up* genre. This is partly due to an attempt by video game developers, struggling with innovative 3D production technology, such as camera control, hit detection, and animation systems that were too clunky for the fast action *Beat 'em up* genre. Thus, little progress was made in game play mechanics and development, the genre stagnated, and became dissected into various other genres, including the 3D action hack and slash, and the isometric action RPG. More recently, these two genres have overshadowed the *Beat 'em up* genre so that currently, the genre is either little understood, or completely unknown.

Thesis Statement:

> We demonstrate that Beat Em Up production is feasible with a small team of developers using 3D technology.

1. We examine the production issues and problems that arise in porting the Beat Em Up genre from 2D to 3D.
2. Most of these problems focus around production issues
3. Since concept art provides the foundation for a managable production schedule, we describe an ideology that realistically informs goals that are feasible for a small production team.
4. We translate this ideology into palpable concept art that demonstrates the visual simplicity required for the Bean Em Up game that we develop.
5. As a proof of concept, we have developed the concept art and the 3D assets required to build a 3D Beat Em Up video game.
6. We present a game level that harkens back to the popular mechanics of the 2D Beat Em Up genre, and show their adaptation to 3D technology.

In this paper, we address some of the technological issues and shortcomings that led to the recession in popularity of the *Beat 'em up* video game genre. In particular, we adapt current technological advances and apply them to 3D video game development to mirror the mechanics of the 2D *Beat 'em up* genre. We describe our assembly-line approach to 3D asset production that permits quick development of multiple and varied characters and environments. Our approach to 3D camera placement provides the same viewpoint used in the 2D game approach. We have developed various scripts to streamline the animation process. In addition, levels are designed retroactively with a "go right" mentality, and various programming techniques that facilitate swappable, and multiple enemy types with different AI behaviors. Finally, we apply current hit detection technology to the spirit of the *Beat 'em up*.

The rest of this thesis is organized as follows. In the next chapter, we provide background information about the terms and concepts that we use in this thesis. In Chapter 3 we describe the history of the beat em up genre and describe some canonical games in this genre. In Chapter 4, we describe our methodology, and in Chapter 5 we describe the video game that we have developed using our techniques. Finally, in Chapter 6 we draw some conclusions.

Chapter 2

Background

In this chapter, we describe the fundamental terms and concepts that comprise the Beat-Em-Up, or *Brawler*[1]. The Beat-Em-Up is similar to many of the *2D action games* of the period, such as *tournament fighter* and the *action platformer*. Thus, the Beat-Em-Up is easily confused with these other genres that were popular at the same time. The action platformer and the tournament fighter are similar to the Beat-Em-Up, yet different in some significant aspects. The action platformer relies on precision jumping and very basic attacks where the player cannot walk vertically and is constrained to a strict two dimensional plane. The tournament fighter differs because the action can be described as *intimate*, in that the players attention is focused on a single opponent, whereas in the action platformer the player must confront multiple enemies and obstacles in parallel with evasive actions as part of the combat. In both cases, the players are constrained to 2d planes with the ability to crouch and jump. The defining properties of the Beat-Em-Up include moving both horizontally and vertically, defeating multiple combatants from multiple directions, minimal platforming and puzzle elements, cooperative play among 2-6 players, and a varied move set of attacks the player can chain together. The player's objective is to advance through a series of stages each with various settings and themes. The player moves horizontally through the level and is forced to stop and fight multiple enemies to *continue* advancing. At the end of each level, the player is confronted with a climatic fight with a "boss' enemy that possesses considerably more strength than the normal enemy.

Through the late 1970's to the mid 1980's, video arcades were the most prolific platform for video game development. It is through this medium that the Beat-Em-

[1] In this thesis, we use Beat-Em-Up and Brawler interchangeably

Up surfaced. Many shooting and platformer games had been developed, but two player fighting games were the catalyst for the Brawler's evolution. With the allure of easy to learn mechanics, cooperative gameplay, and simply designed yet difficult level progression, Beat-Em-Ups became a hit among arcade venues. Japanese developers can be credited with forming and popularizing this new genre of game using sources from martial arts tournaments and themes.

The gameplay mechanics of any video game define the identity of the genre, and this identity becomes obvious when game samples from each genre are juxtaposed. Gameplay influences the game's narrative, programming, art assets, and the overall classification for which the game will be identified. Like all video game types, the Beat-Em-Up follows a very strict set of mechanics that it must adhere to in order to be accepted as a member of the genre. Gameplay systems can be partitioned into three main categories: the player's interaction with their character, enemy artificial intelligence (AI), and gameplay design.

Video games are an intriguing medium in that they allow for the audience to be immersed in the work through real time interaction with the story line's main protagonist(s). In order to achieve this desired effect, the game must have systems in place at a fundamental level so the player and the character they control can communicate with one another. Beat-Em-Ups, while a simple genre, have very specific mechanics that inform this conversation. The player's movement, ability to attack and be attacked, and ultimately the success or failure of the player are defined by a game designer and implemented by its programmer.

Player movement in a Brawler is denoted by the ability to not only traverse in the horizontal direction, but also in a vertical one. This does not mean the player can walk straight up into the air, nor can he pass downward through the ground. Rather, the player can move upward to progress deeper into the back ground of the playing field, and move downward to obtain a position closer to the foreground. Perspective and camera placement take paramount importance in creating this illusion. A set background is placed at the middle of the viewing screen, usually with multiple layers using parallax scrolling to create depth. The floor or ground serve as a playing area, and travels downward creating a foreshortening effect in which depth is implied. In this way, a player can move in eight individual directions around the playing area, but ultimately may only face along the horizontal plane. If the player is facing right and decides to walk upwards/into the playing area, he still faces in the right direction.

Free range of movement is a characteristic of the genre, but the player may only attack in the horizontal direction. While this may seem like a hindrance, attacking has a forgiving hit detection system in place, allowing for multiple enemies to receive damage along a generous vertical range. Attacking also incorporates a stun effect upon foes. The first hit will interrupt an enemy's action, and allow a chain of attacks to *continue* until the enemy is defeated or knocked down. These two attack mechanics are very important to fair gameplay due to the presence of many enemies attacking from multiple positions.

During the evolution of the Beat-Em-Up, various new attack types have been implemented. The earliest of Brawlers debuted only simplistic punching and kicking. Over the years, jumping, attacking while jumping, grab attacks, grab throws, 360 attacks, backward attacks, and special moves were incorporated into game play. The ability to attack while jumping effectively enabled the player and his avatar to move around the playing field with a greater sense of mobility and attack options. Grabbing became a stable in the series, allowing the player to move close to an enemy and latch on to them. This state opened a whole new set of options for attack. The player could chain a multitude of attacks at very close range, or throw the enemy, knocking all others to the ground if they were in the way of the catapulting foe. It is inevitable that the player will be surrounded by enemies and in these situations, the 360 degrees attack knocked down all enemies to the ground near the protagonist. This attack and other special moves usually cost the player resources such as health, and as such, balanced these all power moves. These other special moves varied between game, but all shared common characteristics. They were more effective than standard attacks; they affected multiple enemies, and cost the player resources. Such resources include health, a recharging *special bar*, or collecting various power-up items on the playing field.

To add tension to the playing experience, the player character can and will be attacked and eventually will be defeated. The major difference between the player and enemies, pertaining to defeat, lies within their respective health bars. The health bar is a visual representation that informs the player how many more attacks they can sustain before they are defeated. While enemies have a health bar as well, their meter will vary in length, sometimes being very short or much larger than the player's. Each time a successful attack collides with the player's hit detection box, a portion of the health bar depletes. Enemies have unique attack characteristics from one type to another, and in this way, an element variation is introduced to the player. When the player's health bar is emptied, they are defeated.

There are usually several tiers of continuation for the player character, the base level being the aforementioned health bar. The player also possesses *lives* that are represented by a number tally. When the health bar is emptied, the player reappears on the screen with full health, and their *lives* tally is reduced by one. Extra *lives* can be earned though. As with most games from the arcade era, a score meter indicates how well the player character is performing. Defeating enemies, picking up items on the player area, and clearing levels are all ways that increase the score. Reaching certain levels on the score count will award the player an extra life.

Some Beat-Em-Ups also have items that increase the player's *lives* tally when the item is attained. Total defeat comes when a player's *lives* tally reaches or exceeds zero. What happens afterward diverges when examining the different platforms of the Arcade and the Console. An arcade machine will display a countdown asking the player to insert money into the machine. Doing so will award the player with more *lives*. Consoles are different in that they have yet another tier of continuation called the *continue* Typically *continues* cannot be attained and are shared among all players.

Programed, artificially controlled enemies share some of the same characteristics as the player. They use the same health bar systems, attack and hit detection, and movement restraints. While no human player actively directs these enemies, a series of programmable scripts dictate how these characters behave. It is a delicate balance to control how aggressively the artificial intelligence attacks the players. Typically it is the goal of the programmer to create a behavior that has the enemies circle randomly around the player characters, and attack at times that provide the desired amount of difficulty. Again, creating unique degrees of these behaviors creates more variation for the player to experience in the gameplay. These non-playable characters can appear on the playing field in numerous ways. They can come horizontally from non-visible space off screen, or various rooms and doors. One aspect always remains true though; when enemies appear, the player must defeat them all to progress through the level. An enemy is defeated in much the same way the player is. When the foe's health bar is depleted, they are effectively dealt with and fade from the viewing screen. The main difference between the player and the enemy in this regard is that an enemy has no extra *lives* or *continues* The overall gameplay of the Beat-Em-Up is somewhat straightforward. A player character (or multiple player characters working cooperatively) proceeds through a linear path with the objective to reach the end. A fixed camera follows the player at their profile displaying a fairly large view of the playing area. The camera stops moving and locks the player in a section of the level in which he fights multiple enemies. The camera does not unlock until a pre-defined amount of enemies have been defeated. At this time, the player once again moves closer to the end of the level.

Throughout the level the player can interact with various objects in the world such as destructible furniture, health bonuses, and useable weapons. It is rare for the player to interact with the environment in ways that allow access to the rest of the level, and if these situations do occur, it is usually in the form of physically destroying an obstacle. An archetypal Beat-Em-Up is usually set in an underworld or urban environment with revenge motifs as the main impetus for the protagonist. An overall theme for the setting is established, and each level offers new locals within this world. When a player completes one level, he is translocated to the next with very little in the way of transition in between. Players win the game when all levels have been completed without losing all of their *lives* However, if a player exhausts the cache of *lives* he possesses, he fails and must start the game over from the beginning. Throughout the game, difficulty is scaled so that the player faces increasing challenge as he progressed through the levels. This is accomplished by adjusting properties of the enemies such as their aggressiveness, their health amount, their damage output, and how many enemies appear on the playing field simultaneously.

Chapter 3

Timeline: The Rise and Fall of the Beat-em-up Genre

In this chapter, we review the beat-em-up video games that relate to our work. We discuss the genesis of the genre and the important games that promoted their development and popularity, including the mechanics, graphic techniques, and presentation that were prevalent in 2D technology. We also describe the refinement of the genre from the crude, stiff mechanics of the late 1980's beat-em-ups, to the 16-bit sophisticated, graphical and processing improvements of the late 1990's. We also discuss the maturation of the genre so that larger budgets, programming teams, and marketing talent, together with the growth mandated by a competitive environment, lead to the pinnacle of the genre in the mid-1990s. Finally, we discuss the factors that lead to the decline of the beat-em-up genre, largely credited to its inability to adapt to 3D technology.

3.1 The Genesis of the Melee Combat Game

In the late 70's and early 80's, arcade cabinets were the eminent video game platform. During this time, space shooters and simple platformers were prevalent and popular. A few games however based gameplay on melee combat. A handful of simple boxing and martial arts games let players take combat into their own hands. Early beat em up games were almost exclusively arcade games, and they took their inspiration from the popularity of martial Arts movies that were playing at the time [3]. The first hand to hand combat With Heavy Weight Champ, Sega had started a trend. The game company began as a slot machine and jukebox company in 1940, while later getting into photo booths. The 1970Šs saw an explo-

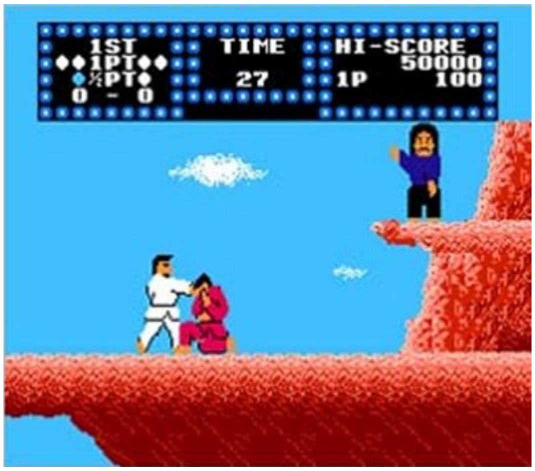


Figure 3.1: **Karate Champ, 1984, Arcade**.

sion of success for the company due to the popularity of arcade gaming platforms (http://en.wikipedia.org/wiki/Sega#Company_Origins_.281940.E2.80.931982.29). Unfortunately The only lasting contribution the boxing game provided for its successors was the fact that it shared the same concept of melee combat.It was not until the middle of the 80's that hand to hand combat become popular. In 1984, Technos Japan developed the first melee combat game that allowed 2 players to engage in combat against each other. The game was called Karate Champ, shown in Figure 3.1, published by Data East.

Karate Champ was a martial arts game, tournament based, with a simplistic plot.

> Like most arcade games, story line, plot and game play are minimal in order to make the games accessible to a wide audience. The primary goal of arcade games was to get as many people as possible interested enough in a game to start putting quarters into the machine and playing it [1].

Players won rounds by hitting their opponent. This game paved the way for 2 players to experience the hand to hand combat incorporated into Beat Em Ups.

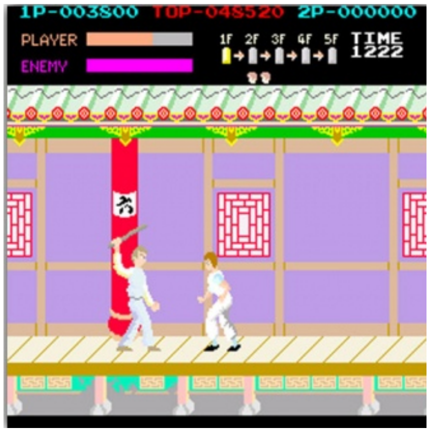


Figure 3.2: **Kung Fu Master, 1984, Arcade**.

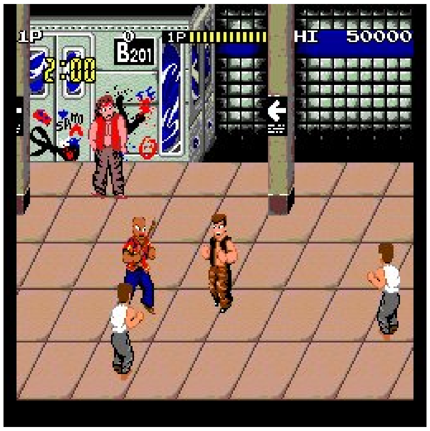


Figure 3.3: **Renegade, 1986, Arcade**.

In the same year that Karate Champ was popular, Kung Fu Master, shown in Figure 3.2, was released for arcades. Irem developed and published the game that introduced the first instance of simple 2d side scrolling levels that incorporated the fixed position camera that Beat Em Ups would employ for years to come. The game's mechanics involve using punches and kicks to defeat enemies as they rush toward the player. The beat em up genre really began in the eighties. The earliest beat em ups owe a lot to early Hong Kong cinema as seen in the Kung Fu Master arcade game [1]. With these two games, the blueprints where laid bare for Beat Em Ups to follow.

3.2 The Emergence and Development of the Beat-em-up Game

With the release of the 1986 game Renegade, illustrated in Figure 3.3, the first true Beat Em Up was born. Technos Japan originally called the game Nekketsu K?ha Kunio-kun (”Hot Blooded Tough Guy Kunio”), a side-scrolling beat-em-up released in 1986 about a high school student who fought thugs and delinquents from other schools and, was the company’s first big hit in Japan. For a company that started out in a one room apartment, much success was to follow. (http://en.wikipedia.org/wiki/Techn

The game was simple, featuring four fighting areas that players were locked into as waves of enemies were thrown at them. It was the first game that allowed players to move not only side to side in the environment, but up and down as well. Renegade also popularized the gritty, urban settings and revenge style story found in many Beat Em Ups. The player defends himself from the underworld thugs and uses street fighting style attacks. A jump function was also incorporated adding a new layer of mobility and giving the player more freedom of choice in how to defeat his opponents. The format of the art style is also a theme that is continued throughout the Beat Em Up genre. By skewing the perspective of the play area, players felt like they really were moving deeper into the environment while they were only moving upwards on the play screen.

The spiritual successor to Renegade is considered by many the grandfather of all Beat Em Ups: Double Dragon was released in 1987, one year after the appearance of Renegade. Technos Japan built upon this very young gameplay formula and crafted a more complete game. Progression by traveling in the horizontal right direction was first introduced along with many more refinements to gameplay. The biggest innovation, however, was the first instance of two player cooperative play. Players could now fight together for the common goal of beating several levels and ultimately the entire game. Another innovation came with giving the player more

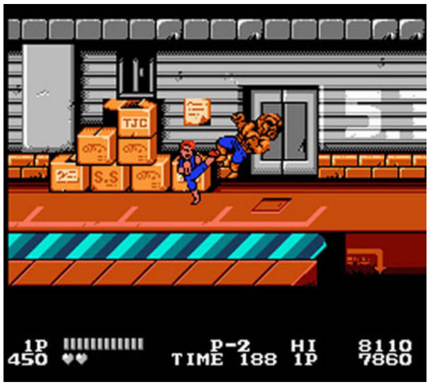
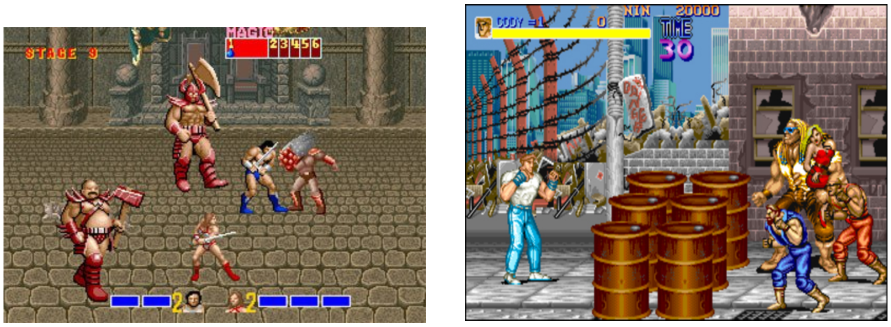


Figure 3.4: **Double Dragon, 1987, Arcade**.

choice in battle. Double Dragon, illustrated in Figure 3.4, was the first game that allowed players to not only use punches and kicks, but special moves and weapons. With the explosive success of Double Dragon, the Beat Em Up was allowed to grow and mature over the next several years.



(a) **Golden Axe, 1989, Arcade**

(b) **Final Fight, 1989, Arcade**

Figure 3.5: Two Video Games from the Beat 'em up genre.

One such title in 1989 named Golden Axe, illustrated in Figure 3.2, offered the same basic gameplay but offered a few new twists. Gone was the urban setting. In its stead was a fantasy theme where swords and magic replaced street fisticuffs. The idea of the *special move* was further expanded upon, and a fully developed magic system was implemented. Players now had to manage not only a health meter, but a magic meter as well. When a special move was used, the magic meter would deplete. While Double Dragon introduced weapons that could be picked up, Golden

Axe introduced mountable beasts to augment and empower the player.

Another monumental title, launched in the same year as Golden Axe, was Final Fight, illustrated in Figure 3.2 Many critics still hail Capcom's Final Fight to be the best game in the genre. Capcom, whose name is derived from the combination of ŞCapsule ComputerŤ (a term coined by the company to describe the arcade machines) made its start developing electronic games in 1979. The company has seen success in the arcades days of the 80Šs all the way till current times with triple A budgeted console games. As far as game companies are concerned, there are few other publishers or developers that have had the staying power of Capcom. (http://en.wikipedia.org/wiki/Capcom) Final Fight refined the Brawler and became the benchmark for others to follow. The player controls were responsive, the graphics made a leap to 16 megabytes and an expanded color library, the art style featured characters and environments that were more detailed due to the close camera placement, and The story was more sophisticated (but not by much). The game shared a lot of commonality with Double Dragon, but now players were given the choice of three characters, each with their own strengths and weaknesses. The game reached critical success, and a ported copy would arrive on the Super Nintendo console system a few years later. In the next section, we describe how transporting successful arcade Beat Em Ups to the now powerful and popular 16 bit consoles of the Sega Genesis and the Super Nintendo Entertainment System would usher in an era of great achievement for the genre.

3.3 The Golden Era of the Beat-em-up Game

The early 90s served as a launch pad for several popular Beat Em Up intellectual properties. Both original titles and brand name franchises began development during this time. With the rise of the more powerful consoles, more and more Beat Em Ups began migrating into households around the country. The Teenage Mutant Ninja

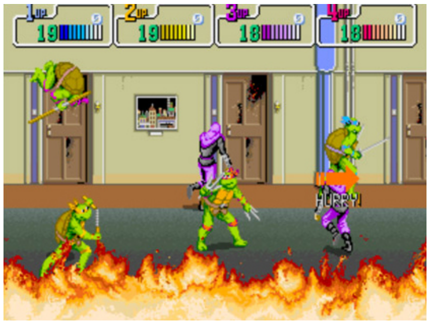


Figure 3.6: **Teenage Mutant Ninja Turtles, Arcade Game**.

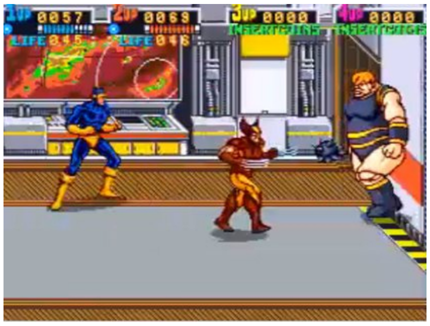


Figure 3.7: **I. Xmen Arcade Game, 1992, Arcade**.

Turtles Arcade Game, illustrated in Figure 3.6, was one such title that became universally popular. TMNT was significant in its use of 4-player co-op gameplay as well as letting players use objects in the environment against their enemies. Teenage Mutant Ninja Turtles, by Konami, was the first to feature 4 players in a pure beat em up, a trend that later followed into many other titles by the same company. The company was founded in 1969 as a jukebox rental and repair business in Osaka, Japan. The name ”Konami” is a conjunction of the names Kagemasa Kozuki (current chairman and president), Yoshinobu Nakama, and Tatsuo Miyasako. Konami is famous for games such as the Metal Gear series, Dance Dance Revolution series, Castlevania series, and Contra series (http://en.wikipedia.org/wiki/Konami). The instant recognition of the licensed properties that Konami used also made their games very popular like the Simpsons arcade game in 1991.

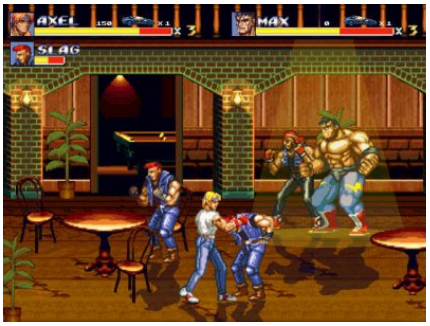


Figure 3.8: **Streets of Rage II, 1992, Sega Genesis**.

The X-men Arcade Game in 1992 took Konami's franchise treatment to another level with the inclusion of 6 player cooperative combat. Full video screens were required to handle all the action, which was another first for the Genre. In a direct attempt to counter the Super Nintendo's alliance with Capcom, Sega created another critically and historically adored title in 1991. The Streets of Rage series produced 3 games over the course of 3 years, but it was the second in the series that many still consider better than Final Fight, and all previous Beat Em Ups: Streets of Rage II. This second game in the series of three games became Sega's flagship Beat Em Up, and offered the same smooth controls, sophisticated graphics, and varied selection of playable characters, but separated itself with an amazing soundtrack and audio effects. This popular title revamped the *special move* system, which allowed players to strategically use several abilities in any given situation. This also separated the character from one another, and offered a lot of replayablity to the series. "While the first Streets of Rage provided 3 different characters to choose from and a moderately lengthened game, it was the second game in the series that set a new bar. Offering a full 8 levels chock full of a variety of backdrops and bosses, this wasn't a game you could sit down and beat in 20 minutes, which was typically the case with the previously discussed titles. It was a game designed for the consoles."

The Final Fight series responded with 2 sequels, both exclusive to, and developed for, the Super Nintendo. The second title's only departure from its series formula

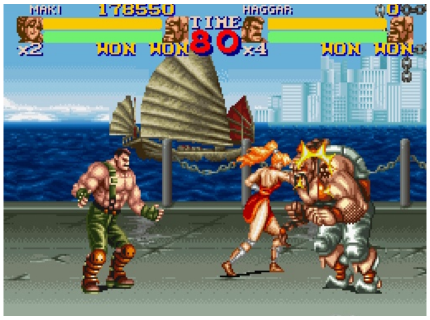


Figure 3.9: **Final Fight II, 1993, Super NES**.

was the setting of each of its levels, illustrated in Figure 3.9. Each stage carried the player to a different country or locale from around the world. Instead of just pummeling thugs in a single city, you could now exchange cultures in several different ones. Final Fight III brought the fight back to Metro City, its original setting. By 1995, developers had unlocked the power of the Super Nintendo, and this allowed several new features to be included in this title. The first Final Fight ported from the Arcade was severely limited and featured only 2 characters instead of the original three, lacked cooperative play, and excluded a whole stage. The third entry boasted 4 characters, 2 player cooperative mode that could substitute a computer partner if no friends were around, and several stages with branching paths one could take, altering the course of the game. Final Fight III also implemented Streets of Rage II's special move system. Special moves now required intricate button presses to be performed.

Other notable Beat Em Ups that were released during this time period include Rare's Battletoads in 1991. BattleToadsăwas always hailed for its incredible degree of difficulty, and certain levels in the game still incite rage in players. BattleToads was so popular that it actually began to cross over with other Beat Em Up games. Since they were sequels, BattleToads and Double Dragon were released on both the NES and Genesis. The Death and Return of Superman, Batman Returns, and Maximum Carnage feature comic book super heroes, and all games are considered successes in the genre. The last respectable Beat Em Up from this period appeared on the Sega Saturn in 1996, Sega's more powerful successor console to the Genesis. Sega created Guardian Heroes with several innovations in an effort to keep the genre from stagnating. The play field was narrowed to 3 different 2-d planes, several characters were featured each with their own unique play styles that could be improved between levels, the art assets where incredibly detailed due to the power of the Sega Saturn, and the story proved to be the best in the genre to date.

3.4 The Emergence of 3D Technology and the Fall of the Beat-em-up Game

The mid 1990's beaconed the emergence of a new generation of home video game consoles capable of rendering true 3d graphics. Platforms such as the Sega Saturn, the Sony Playstation, and the Nintendo 64 became fixtures in homes across America, and the demand for 3d graphics had never been so intense. Genres such as 3d platformers, Japanese RPGs, and First Person Shooters were becoming the dominate style of game. However, the beat em up genre began to decline in popularity during the late nineties as interest in arcades and 16-bit consoles dwindled as well. "It can be said that the 32-bit era and the move to 3D technology in the video game industry officially ended the Golden Era of the beat em up genre [1]." This trend can be supported when looking at the batch of Beat Em Up titles released during this time of change.

Fighting Force, illustrated in Figure 3.10, is a perfect example of what went wrong with the Beat Em Up genre. Released in 1997, this game was developed under the assumption that this would be a 3D sequel to the Streets of Rage series. But a dispute with Eidos Interactive and Sega arose when the topic of porting to other consoles became an issue. Core's Fighting Force for the Playstation received a lot of hype but was not well received upon its release. The studio was part of

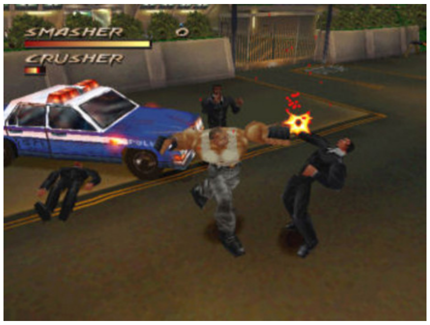


Figure 3.10: **Fighting Force, 1997, PS1/N64**.

distribution company CentreGold when it was acquired by Eidos Interactive in 1996. Eidos subsequently sold most of CentreGold, but retained U.S. Gold, the owners of Core Design. Core had a brief history of producing titles for the Sega consoles, such as Thunderhawk for the Mega-CD and later the original Tomb Raider game for the Sega Saturn [5]. There were many Beat Em Up mechanics that did not transition well to a 3d perspective. The art of this time suffered due to the infantile state of 3d art asset development. This obstacle far too often led to uninspired level design and the reuse of generic enemy skins.

Also Fighting Force suffered from hit detection issues as well, mainly with the player attempting to land attacks on enemies. In 2D, player attacks killed everything in a 180-degree radius. In 3D, player attacks hit a tiny hand-sized section of the world [2]. And finally, like in most games using 3d art assets, camera control issues impaired game play. In 2d perspective, the camera placement and control became an iconic representation of the Beat Em Up. The player would simply *Go Right* to progress through a level. Now, perspective allowed the player to view any angle of the action at hand. Many developers struggled with this issue as common problems included foreground objects impairing the player's view, the camera not centering on the important occurring action, and nausea inducing camera movement. Core's Fighting Force for the Playstation received a lot of hype but was not well received upon its release.

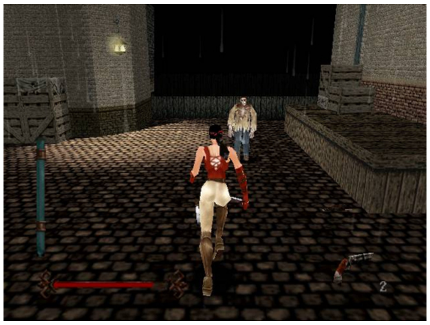


Figure 3.11: **Nightmare Creatures, 1997, PS1/N64/Windows**.

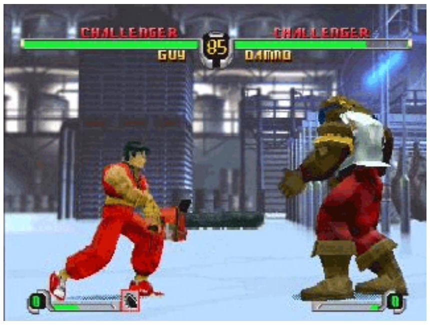


Figure 3.12: **Final Fight Revenge, 1997, Arcade**.

Though Fighting Force wasn't a complete failure, fans and critics alike could only assume the genre would get better over time in its new 3d world. There were some aspects that this iconic 3d Beat Em Up handled with grace and elegance, but later games would prove that developers didn't know how to handle the Beat Em Up in three dimensions. One such game was Nightmare Creatures, illustrated in Figure 3.11, released in the same year.

Activision published a game that did more harm than good to the genre. The combat was stale and unresponsive, camera issues still plagued the genre, and levels were empty mazes that lacked any kind of appeal. A sequel was released a few years later, but did little to improve upon itself. Several more flops would plague the series on into the new millennium. A new generation of more powerful consoles had been released, but still, Beat Em Ups never improved. Even a classic series such as

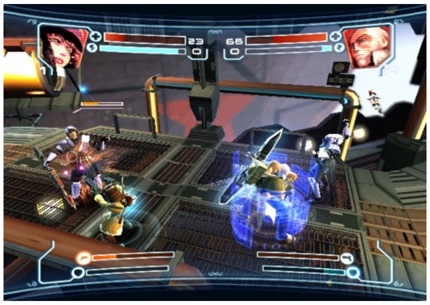


Figure 3.13: **Red Star, 2007, PS2**.

Final Fight floundered more and more with each successive version.

Final Fight Revenge, illustrated in Figure 3.12, dumped the genre it helped make famous, and it became a tournament fighter. Though it was attempted to incorporate 3d technology into the series, it baffled its fan base by its sudden changes, and it was an unsuccessful flop. Capcom tried taking Final Fight back to its Beat Em Up roots almost 7 years later on the new Microsoft Xbox and Sony's Playstation 2. Final Fight Streetwise was supposedly hailed as a reboot to the series, but by 2006, Beat Em Ups had become archaic by design. This was to be Final Fights last game ever developed.

However, two gems did shine through this dark time for the Beat Em Up genre. Acclaim studios realeased a 3d Beat Em Up that borrowed the storyline and setting of a popular graphic novel. The name of the company was picked because it had to be alphabetically above the co-founder's former place of employment, Activision, and also had to be alphabetically above Accolade (another company formed by ex-Activision employees). This was a common formula for picking names of new companies that were founded by ex-Activision employees (the founders of Activision used this formula when they left Atari). The Red Star, illustrated in Figure 3.13, was a game, originally slated for a 2004 release, that didn't see the light of day until 2007 when it appeared for the PlayStation 2. It is one of the last truly exceptional co-op beat em' ups that did something new by combining gameplay from top down

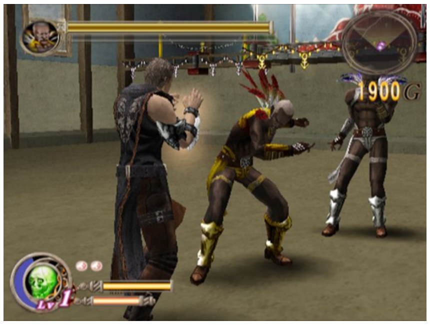


Figure 3.14: **God Hand, 2006, PS2**.

Figure 3.15: **Devil May Cry, 2001, PS2**.

shooters with traditional beat em up action. The camera issues were eliminated by mimicking the 2d placement of earlier games in the genre. Players were able to advance their characters powers over the course of the game as well as adding a degree of longevity expected in games of this period.

Capcom released God Hand, illustrated in Figure 3.14, a year after giving up on Final Fight. This game garnered a decent level of respect due to its incorporation of fist based melee combat and the fast paced action of many 3d Hack and Slash games during this time. In fact, God Hand could be viewed as an example of the evolution of the Beat Em Up. Many critics still consider this game as part of the genre, but so many elements had been extracted and rebuilt that the game resembled the 3d Action games that rose from the ashes of the all-but-dead 3d Beat Em Ups.

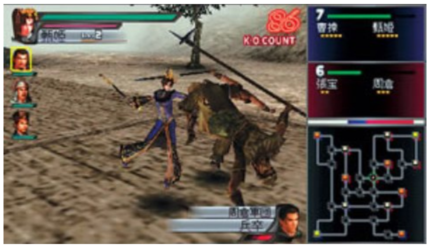


Figure 3.16: **Dynasty Warriors, 1997, PS1**.

3.5 The Splintering of the Beat-em-up into Multiple Genre

As in the case of God Hand, The Beat Em Up's iconic mechanics were stripped and incorporated into other genres in the early 2000's. After the move to 3-D, the majority of games in the genre would eventually merge with action adventure games and much of the action was diluted with the inclusion of other genre elements like platforming, puzzle solving, item fetch quests and quick time event exercises. D_2 A classic example of this evolution is the game Devil May Cry, illustrated in Figure 3.15. Capcom had decided that instead of forcing the Beat Em Up into the 3d world, it would mold, reforge, and modernize the genre. Like many beat em ups, Devil May Cry placed a heavy emphasis on combat versus numerous enemies, all taking place within a gothic setting as players took control of a part human, part demon character known as *Dante.* The game had a deep and complex combo system and fast paced gameplay with highly responsive controls. This game heralded a new genre that can be seen as one of the spiritual successors to The Beat Em Up. The 3d action/3d hack and slash genre would go on to spawn many great games in the new millennium.

Two other titles that followed the Devil May Cry formula include Ninja Gaiden for the Xbox, and God of War for the Playstation 2. Both of these games focused on combat versus multiple enemies, light puzzle elements, platforming, and fast paced, combo driven gameplay. There was another 3d title that redefined and evolved the Beat Em Up using a different direction.

Koei's Dynasty Warriors 2 tossed aside the one-on-one fighting game formula of its first entry. This installment was an interesting title that featured a massive number of onscreen enemies and open battlefields. It also featured some light strategy and rpg elements which made for an interesting combination. It certainly was a unique title in the beat em up genre. However, it was the next game that would truly bring Koei success. 2001 was a significant year for the genre. Dynasty Warriors 3 made some improvements on the formula presented in its predecessor. One notable change was the addition of a co-op feature. Dynasty Warriors 3, illustrated in Figure 3.16, was a big seller in both North America and Japan.

These games addressed and repaired a lot of the issues inherent in previous 3d Beat Em Ups. Camera Issues were brought under control, hit detection functioned properly, and these 3D Beat Em Ups were just fun to play again. Nicholas Puleo, an online journalists, said it best when he proclaimed, "An evolution of this genre is not ideal, but we'll take the co-op when we can get it. I know I'd love to see some brand new next-gen beat em ups. I want classic, down to the streets side scrolling style with some great co-op play. There's no reason the genre all of us grew up loving can't make a comeback [4]."

3.6 The Neo-Renaissance of the Beat-em-up Game

Toward the end of the 2000's, the traditional Beat Em Up started making a comeback. This can be mainly attributed to the advent of downloadable distribution in which the consumer can pay, download, and play a game solely online. This payment and distribution plan cuts out a lot of overhead in terms of mass production and market retail costs. The result is the ability to produce games on extremely low budgets, and then sell these games for comparatively low prices. With the rise of

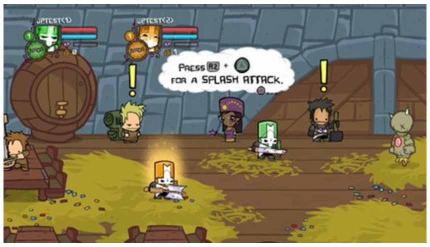


Figure 3.17: **Castle Crashers, 2008, Xbox Live Arcade**.

production costs for modern games, these bargain prices have created new markets for the gaming population to take advantage of. The importance of downloadable distribution has been immeasurably important for the Neo-Renaissance of the Beat Em Up. This is part due to the simplistic nature of the Beat Em Up itself. In the current video game industry of bloated budgets and ever expanding game content, the 30 to 50 minute play-through of the Beat Em Up just doesn't warrant the financial backing it used to garner in the early 90's. The industry is seeing the rise of old school genres due to their simplicity and cheap production costs.

New titles such as Castle Crashers, illustrated in Figure 3.17, from The Behemoth have found very lucrative homes on Xbox Live Arcade. If the popularity of a title like this is any indication, people miss their 2d co-op beat em ups. The Behemoth is a video game development company that was created in 2003 by John Baez, artist Dan Paladin (aka Synj), and programmers Tom Fulp (the creator of the popular Newgrounds Flash website). The Behemoth development studio is located in San Diego, California. The company is known for producing simple games with Paladin's signature 2D style. Their first console game, Alien Hominid, gained critical acclaim by the media and the members of The Behemoth quickly gained status as indie developers focused on bringing old-school styles of video games back into mainstream gaming. games like Golden Axe in that it featured a medieval fantasy theme with magic systems. The amount of characters playable is also an amazing achievement. While initially there were only 5 different knights to play as, downloadable support

Figure 3.18: **Scott Pilgrim vs. The World, 2010, Xbox Live Arcade**.

has steadily funneled new playable characters, each with their own unique magic attacks, into the game. A level up system and loot collected was borrowed from the RPG genre which also added to its mass appeal. The colorful, spirited brawler Castle Crashers hit 2 million players on Xbox Live as of Christmas, developer The Behemoth announced last week, claiming the game is "among the first" XBLA titles to hit the milestone. Another title to harken back to Beat Em Up games of the 90's is Scott Pilgrim vs. The World.

Based off the graphic novel of the same name, Scott Pilgrim, illustrated in Figure 3.18, and released beside the debut of the Motion Picture, This 2d Beat Em Up has introduced new audiences to the *Go Right* gameplay that may not have grown up in the golden age of the genre. Again, a persistent level up system is present in the game as well as currency gained from defeating enemies that allows you to buy power ups for your character. The action feels like it was stolen from a 16 bit console, and this feels right at home in the game. Not only have there been new intellectual properties created during this outburst of Beat Em Ups. The aforementioned classics have also been rereleased as downloadable games for a very reduced cost. Games like Final Fight, Streets of Rage 2, Guardian Heroes, and The Xmen Arcade Game have seen this treatment. Even Konami's classic Turtles in Time was reproduced and given a complete 3d model overhaul. The game was released on Xbox Live Arcade under the title Turtles in Time ReShelled. The financial success of these releases only proves that in the right environment and using the right distribution methods the Beat Em Up is still relevant in the landscape of modern gaming.

Chapter 4

Improving the Production of the Beat-em-up Genre for Small Teams of Developers

This section will detail the process that is employed in the production and development of a 3d Beat Em Up video game. The artist first describes the issues involved in streamlining the production pipeline for creating a 3d Beat Em Up in a team environment of 2 people. The primary goal is to examine the processes in the current video game pipeline and optimize the workflow so the methodology can become manageable for a small team of developers. The Project begins with the creation of a cohesive art style through concepting, or creation of concept art, for 2d illustrations. Character creation is next, followed by rigging, and animation of the characters that appear in the final product. Environment production then begins, and lastly the art assets are used in the game engine. The sections that follow introduce each production stage in the project pipeline, pose the general problems small teams encounter during this process, and describe the methods used to solve these problems.

4.1 Art Direction and Concepting

Before production of any game can begin, a pre-production process is needed. This includes defining a clear art style for your project, and creating informative 2d concept art that captures the spirit of said art style. This process starts with collecting various reference images that visually communicates the style, mood, and theme for the game. Beginning with these materials, 2d illustrators derive drawings for every art asset required for the game. These drawings range from illustrations that are intended to capture a mood, or loosely inspire other artists to specify

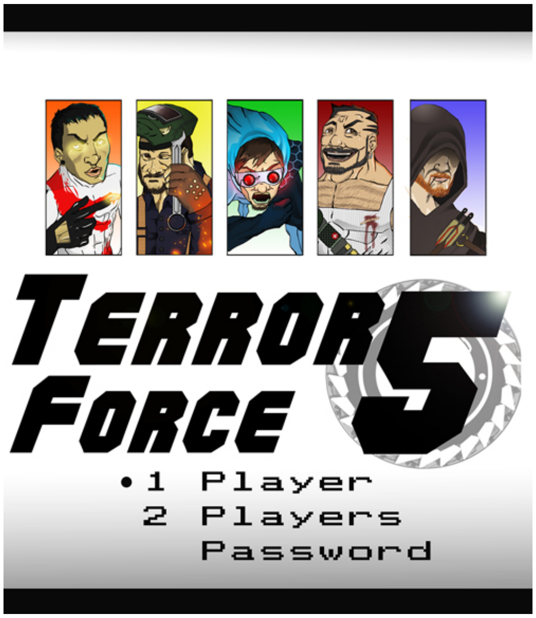


Figure 4.1: **Co-authoring the Art Direction with the concept artist**.

technical drawings that act as blueprints for 3d artists.

Pre-production is incredibly critical to the success of a small team of game developers. This is the planning stage in which all major decisions and design choices are made to form the groundwork of the entire project. A small team must design for a manageable project schedule and pipeline. The development of a Beat Em Up game requires solutions to many interesting design decisions. Since most of the gameplay involves combat against multiple opponents, the art director must decide how to portray several bipedal characters that appear on screen simultaneously. In the past, color pallet swaps and subtle character design modifications allowed the designer to create archetype enemy classes and use these variations to add opponent diversity to the game.

Environments are also a concern considering that Beat Em Ups are always moving throughout a level. Logical yet diverse locales are very important in immersing the player into the world due to Beat Em Ups somewhat repetitive gameplay. In order to ensure a manageable production schedule, The concept artis is closely worked

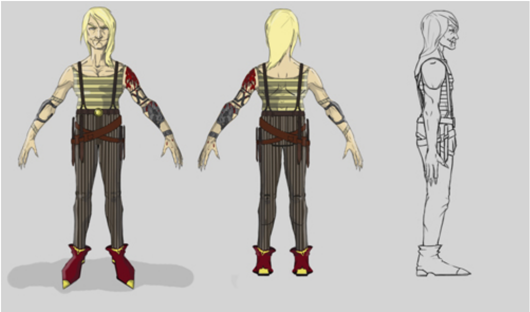


Figure 4.2: **Creating a Standardized Character Concept Sheet**.

with to develop an art style that remains stylistic and appealing yet simplistic. Together, the 3d artist and the concept artist co-authored the art direction. An art style was agreed upon on using influences from the action cartoons of the mid 80's such as G.I. Joe, Thundercats, and the Teenage Age Mutant Ninja Turtles. As for theme, ideas borrowed from super hero comics and the popular TV series "*The A Team.*"

Through this, the creation of a 5 man mercenary group, each possessing his own unique powers and personality was complete. This team of heros operates on renegade type tactics, and are constantly repremanded for their destructive solutions. In essence, the Terror Force 5 fights terror with more terror. This time period was known for having colorful and fantastic characters in every medium, and the Terror Force reflects this. Each character is comprised of simple pen and ink illustration techniques, and relies heavily on flat color. Bright vibrant colors were used to make the character stand out from the backgrounds in order to assist visibly tracking him on screen. Stylistic proportions and facial features were designed not only to adhere to the cartoonish aesthetics, but also to give the 3d artist more freedom in his workflow. Each character was given a unique theme and personality, which in turn drives the visible design of the avatar. The result is a varied cast of playable characters that differentiate themselves in their play mechanics and their visibility on screen.

The setting and theme of the game also remains true to the 80's overtones. The idea of a cold war era antagonist and plot was created. The manical Crimson Klaus, the son of a Nazi facist and a russian prsionar of war, acheives immortality through experimentation of radiation. As a result, he becomes mad and is transformed into a hulking, decaying zombie. The game takes place in an abandoned Russian nuclear power plant. Here, Klaus is determined to laucnh nuclear warheads at both Moscow and Washing in hopes of starting another world war. Numerous photographs of true to life power plants in Russia were referenced, and many propoganda illlustrations were gathered to communicate enemy design. The concept artist used these reference materials to create quick sketches in order to design the characters. The concept artist then created strict character illustrations that the 3d artist could use as blueprints for 3d modeling. The concept artist was instructed to draw a single character in 3 different perspectives, while the front and back perspective include color. A side perspective was drawn only in contours to visibly aid proportion intention. The concept artist created these concept sheets for 2 playable characters, 2 basic enemies, and 1 "boss" enemy, as illustrated in Figure 4.2.

4.2 Character Production: Modeling and Sculpting

With a full library of character concept art to reference, modeling the respective art assets can begin. The current generation of games now support very detailed 3d character models constructed and formed through the use of polygonal faces. There are many software platforms that can create these 3d models, and many are vastly different in their use. Highly detailed models are generally created in 2 different ways: the first begins with a low poly base model, which is used as a base to create high poly models for details. The second method begins with the creation of a highly detailed model, and then building a low poly model using the high poly sculpt. However, the end result is always the same: a low poly model, roughly 3,000

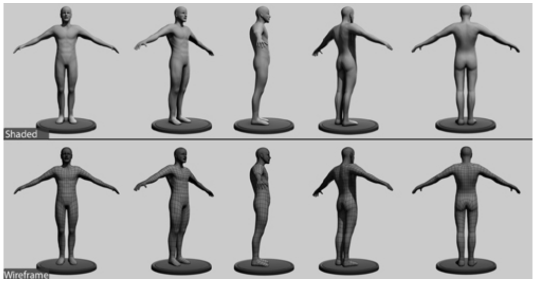


Figure 4.3: **1. Creating a Base Model with Correct Topology** .

to 10,000 quadrangular faces, that captures the contour of the high poly sculpt as best as possible. The high poly sculpt is also required for later use in transferring the fine details to the low poly model.

The modern 3d Brawler poses a few challenges to small/one man art teams in terms of character creation. The basic premise and core gameplay design behind this genre of fisticuffs lie in combat with multiple bipedal characters. A great deal of variation in the visual design of these characters is necessary from visual appeal and a mechanics standpoint. The player would quickly become bored fighting the same foe during an extended play session. Visual flair through unique variations keeps the player interested, and it also allows the designer to distinguish different enemy types from one another. Beat Em Up's old 2d ancestors usually achieved this through color palette swaps and subtle design changes among similar enemies. The end result is the necessity of many unique bipedal characters. This poses a problem to the 3d game world as the character creation process is many times more intricate and resource heavy than simply drawing pixelated sprite sheets. To alleviate this work load issue, efficient and flexible character modeling workflow was created to allow for rapid iteration and quick production turnarounds.

The process began by creating a base bipedal model with correct polygonal topology as shown in Figure 4.3. Topology, generally speaking in 3d modeling terms, is the flow or direction in which polygons are constructed. Deformation of the model during animation relies on the correct placement of these polygonal edges

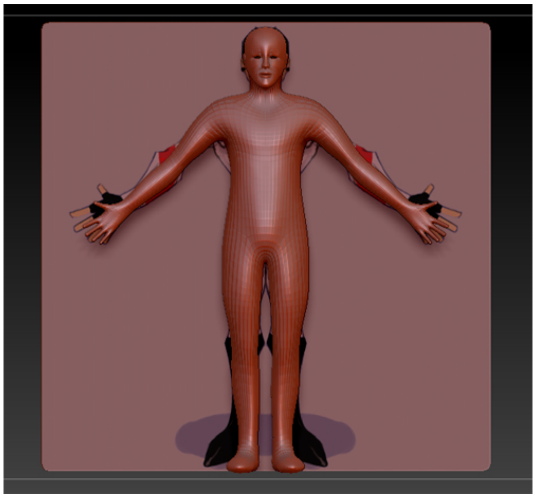


Figure 4.4: **1. Using the base model to begin each character** .

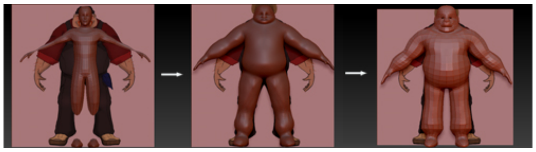


Figure 4.5: **1. Using Zbrush to Import Concept Art into Program to Sculpt Proportions** .

to mimic muscle groups to create realistic range of movements. By creating a base model with the correct topology already in place, the need to re-topologize the low poly game model for each character was eliminated. It is from this base model that each character was modeled from.

Using Zbrush, both the character's concept art, and the base model as seen in Figure are imported into the scene 4.4. The Image Plane tools in Zbrush are used to paint the concept art onto a simple 2d plane. This is done to use as a real time reference during the proportion sculpting process. The base model is then imported into the same Ztool, but on a separate subtool. This effectively separates the concept art and model on separate layers. After the correct assets are in place, proportion sculpting process begins as seen in figure 4.5. Zbrush's move and inflate brushes are used to match the form of the base model to the specific character concept art's

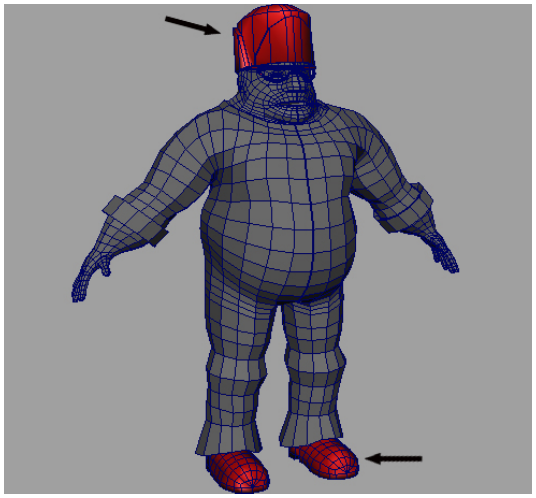


Figure 4.6: **1. Using Maya to Model additional features unique to character. Ex. Hair, accessories .**

shape and size. The "Brute" enemy type is used as an example in order to show the effect of taking a slim model and sculpting it into a larger one, all while using the same poly count.

After the proportions for the character are sculpted, the steps needed to complete the model are assessed. Any extra features unique to the character are then produced. These features include extra polygonal faces extruded from the model to create sleeves for a shirt, extra meshes such as hair, hats, or shoes to be modeled, or eliminating parts of the model that are not necessary such as an amputated arm. The Figure 4.6 highlights in red extra meshes that were added to create the characters hat and shoes. Also notice the extruded polygons around the character's elbows to serve as sleeves. At this point in the workflow, a rough but complete approximation of the character's form has been modeled. The character model should have multiple meshes, and should possess a low poly count. The process of fine detail sculpting can now begin.

Each separate mesh of the character model is exported from Maya as an .obj file, a universal file format for 3d models. Each mesh of the character is then

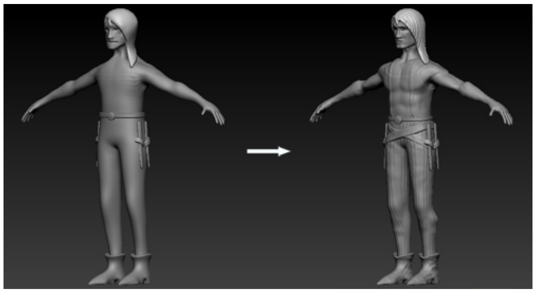


Figure 4.7: **1. Importing Newly Modeled Mesh(es) into Zbrush for high poly sculpting** .

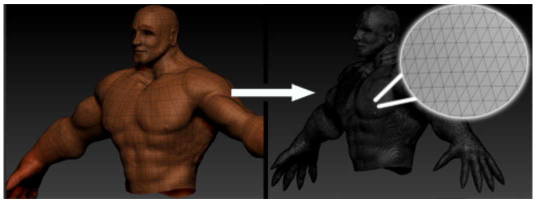


Figure 4.8: **1. Decimating Models and Polypainted textures into a .obj file for use in Topogun** .

imported back into Zbrush as separate subtool layers in order to hide and/or isolate specific meshes. This allows the sculptor to focus on specific parts of the model, without say, the torso getting in the view of the head from certain angles. This also allows the sculptor to subdivide the complete model into more polygons without hurting the performance of the application. From here, the artist uses Zbrush's ability to increase the polycount of a object into the millions allowing for fine detail to be modeled into the surface of the mesh. Several unique brushes and masking techniques are used to create realistic detail. Figure IV B 5 shows the "Acrobat" character before being finely detailed and afterwards as shown in Figure 4.7. During this process, the model is polypainted for use in the texturing stage. This process will be discussed further in the following section.

After the character's high poly sculpt and low poly model have been created, preparations must begin to transfer the fine detail onto the game model. The poly count of each mesh is lowered to its original density and exported as an .obj once more. This allows the capture of the silhouette of the high poly sculpt more completely then if the original and non-sculpted .objs had been used. With the high poly sculpts being comprised of millions of faces, this creates a problem when importing the sculpted mesh into other programs for map baking. A Zbrush algorithm called the Decimation Master was created in order to make these high density sculpts useable outside of Zbrush. Loosely, this is achieved by re-topologizing the high poly sculpt into triangles that efficiently describe the form without the waste of unnecessary faces. Figure 4.8 shows a before and after picture of the torso of the Crimson Klaus figure that highlights the visible result of this process.

Decimation Master can be used to reduce the poly count of sculpts consisting of 10 million polygons down to 2 million, for example, and still retain the same high level of detail. The newly retriangulated sculpts are then exported as .objs to be used during the texture creation process featured in the following section. With this pipeline process, the need to create a base model for each character is eliminated, the need to retriangulate each character is eliminated, use of efficient modeling techniques is best suited to each phase of the modeling process, and the creation of both a high poly sculpt and low poly game model to be used in subsequent applications is fast.

4.3 Character Production: Texturing and Map Baking

With the character models completed, the process of creating various texture maps is next. In today's modern game, texturing is more than just using an image editing program to draw out simple color maps. The advent of normal maps, specular maps, ambient occlusion passes, and emissive or glow maps have created a very technical workflow for the texture artist. Each map has a specific purpose, and when implemented into a game engine shader, they create a product greater than just the sum of their parts. Various and specific steps within the workflow are

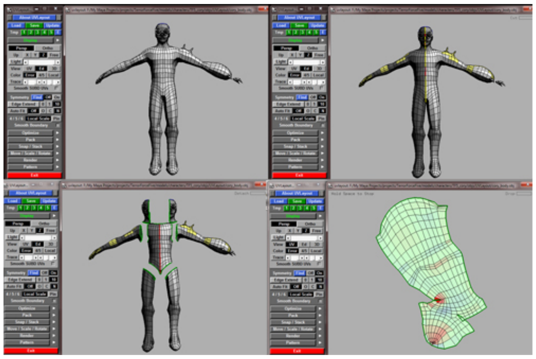


Figure 4.9: **1. Using Headus UV layout to create UV maps. Part 1.**

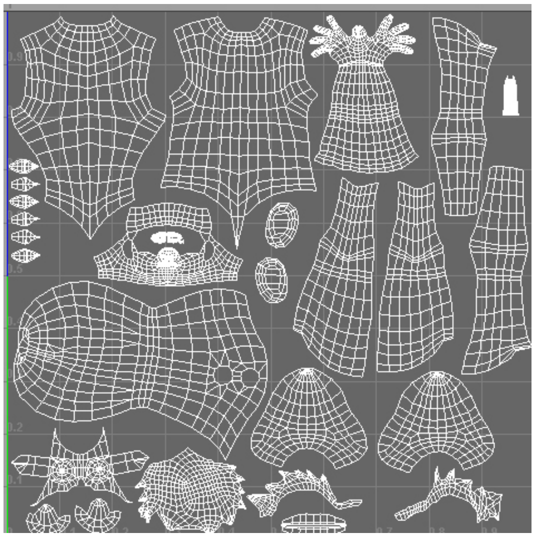


Figure 4.10: **1. Using Headus UV layout to create UV maps. Part 2.**

necessary to perform in the correct sequence in order to produce correct maps.

When texturing for a Beat Em Up, many of the same obstacles that appear in the modeling workflow exist with the texturing process. Many characters requiring many different texture maps compound the amount of work an artist must complete. The act of color swaps on the same model to create even more variation is also the texture artist's job. A quick and efficient workflow must be used to produce several maps for each character.

Before any maps can be baked (This process will be explaind later), each low poly game model must first have their UV maps created. This is the process in which the 3d mesh is cut at logical seams and unwrapped onto a flat normalized surface. This process can be seen in other real world applications such as flattening global maps onto a piece of 2d paper. In order to perform this procedure a program called Headus UV Layout is used. The low poly game model .obj files are imported one at a time into the program. Texture seams must then be designated along polygonal edges. These seams are the location in which the mesh is cut, divided, and laid flat. Each separated piece of the mesh is then flattened using Headus UV Layout algorithms to accurately express the 3d surface space as best as possible on a 2d plane. This process can be viewed in Figure 4.9.

The end result is a UV map with every polygon of the character expressed in two dimensions as seen in Figure 4.10. With the UV map created, the artist can now transfer and create 2d maps that will later be applied to the 3d model. In this workflow, the base color is created as well as the normal bake reference while sculpting the high detail model. During this sculpting process, a technique called Polypainting, a feature of Zbrush, is used that permits the addition of color to the model while it is sculpted. This is accomplished by treating each polygon as its own solid color. When working with millions of polygons, this process can be seen as painting onto pixels, but the pixels are actually small polygonal faces.

When decimating the high poly sculpt into a more manageable poly density, the polypaint details can be transfered as well. At this point in the process, a low poly character model with correct UV coordinates and a high poly sculpt that contains color information from polypainting is finished. The process of baking these details from the high poly mesh to the low poly one is now able to be performed.

The definition of baking, as it pertains to computer graphic production, is the process in which the information from a source containing fine detail is transferred over to another source that is usually more resource friendly. Generally speaking, this is accomplished by importing both the high poly sculpt .obj file into the scene

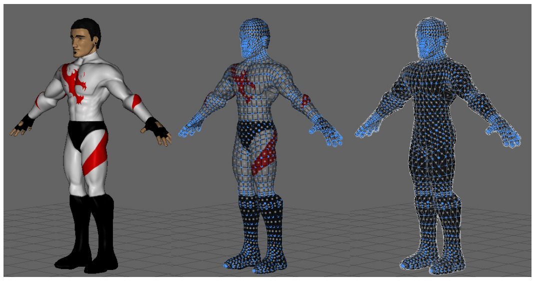
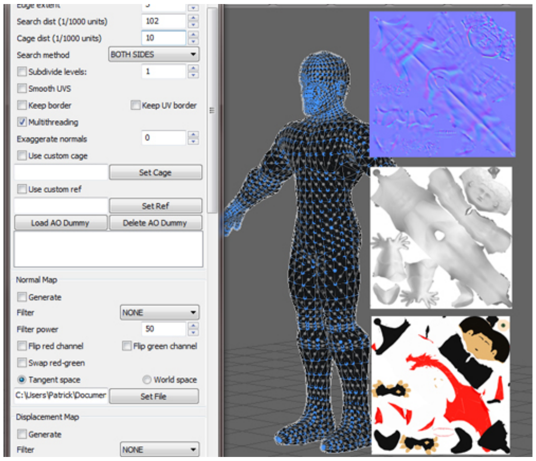


Figure 4.11: **1. Using Topogun to Bake maps. Ex- Ambient Occlusion, Texture, Normal Part 1.**



Search dist (1/1000 units) 102

Cage dist (1/1000 units) 10

Search method BOTH SIDES

□ Subdivide levels: 1

□ Smooth UVS

□ Keep border □ Keep UV border

☑ Multithreading

Exaggerate normals 0

□ Use custom cage

Set Cage

□ Use custom ref

Set Ref

Load AO Dummy Delete AO Dummy

Normal Map

□ Generate

Filter NONE

Filter power 50

□ Flip red channel □ Flip green channel

□ Swap red-green

● Tangent space ○ World space

C:\Users\Patrick\Documer Set File

Displacement Map

□ Generate

Filter NONE

Figure 4.12: **1. Using Topogun to Bake maps. Ex- Ambient Occlusion, Texture, Normal Part 2.**

as a reference. The newly UV low poly .obj file is imported in second, which lies atop the sculpted mesh like a 2nd skin. Topogun then creates a 3rd mesh that acts like a cage. This cage mesh is used to calculate each different map a user requests using a process similar to ray casting rendering. These 3 meshes can be seen in Figure 4.11.

In this example, 3 separate maps are baked using a retopology application named Topogun. Since a high poly character is already painted, the low poly mesh needs to have this color information tranfered to it. This is done by baking. An ambient occlusion map is also calculated using Topogun. Ambient occlusion is a shading method used to simulate accurate shadowing effects that occur on surfaces that are very near one another, such as the shadows in a corner of a wall. An ambient occlusion map is calculated using the high poly sculpt as a reference. For instance, large areas such as the arm pits and under the chin receive this type of shading, as well as fine details such as the subtle slopes a of character's abdominal muscles.

Lastly, a normal map must be created. Normal maps simulate fine surface detail by using RGB color values to simulate the direction of a surface normal. The end product is an inexpensive way of simulating lighting and shadowing details on a low polygonal model. The character's normal maps are derived by using the high poly sculpts using Topogun's normal map baker. These maps can be seen in Figure 4.12. The menu is shown on the left of the figure containing various settings used during the baking process. The right side of the figure shows (from the top down) the characters final normal map, ambient occlusion map, and color map.

Since most characters consist of several separate meshes, the baking process may be performed several times for each character. In order to retain as much detail in each high poly sculpt this is unavoidable. Having a complete model consisting of several million polygons is much too cumbersome to work with, and it may crash the application the artist is working in. Since several parts of a map will be produced for just one character, the artist must use an application such as Photoshop to combine them. This also allows the artist to make any small touchups that the bake process may have messed up on.

By creating the basis of the character's color map using polypaint while sculpting in Zbrush, two processes are combined into a seamless workflow. Tailored programs that quickly and easily preform the tasks of UV layout and map baking are implimented. High fidelity maps are also produced by baking parts of a model one at a time. The whole process is very technical and time consuming, but this workflow creates a seamless process in texture creation.

4.4 Character Production: Rigging

After character art assets are produced, the models must be animated. Before the animation process can begin, however, the technical art of rigging must be completed. Rigging is the act of creating a series of tools that allow animators to pose the characters during animation. Making these handles consist of a very technical sequence of steps, the first being creating a skeleton for the character. Special tools and bones are then created to serve any specialized action for the particular character. Selectable handles are then logically placed near their respective joints and bound to their respective bones. With the skeleton and control handles in place, the skeleton must be bound to the model. The rigging artist must then edit the way the bones deform the model. This process is called weight painting, and involves setting values for each vertex and how the bones near them affect their deformation.

Rigging is a very time consuming process. With so many characters in a Brawler, pre-planning is essential in the pre-production phase to avoid huge time expense. Certain production methods must be implemented in order to streamline the process. Color palate swaps in the texture phase are the most simple, in that they do not affect the rigging process at all. Subtle design tweaks can be accomplished in the sculpting and texturing phase as well, as long as any polygonal faces are added or removed. Extra props and accessories can be added to an existing rig as well with little extra overhead work involved. All these aforementioned techniques assume that the artist is only iterating on a common enemy type. There will always be vastly different character models to rig, and for this an automatic rigging system is created that eliminates production costs by half of the artist's time.

To create an auto rig system in Maya, the mel scripting language native to the application is implimented. Maya has a very intuitive scripting workflow, in that every action the artist preforms in the platform, a window displays his actions in mel format. In effect, the arist would rig the character manually, and for each step

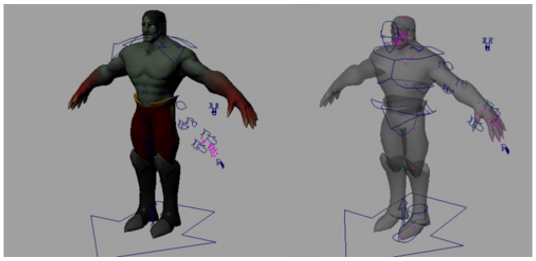


Figure 4.13: **1. Creation of an Auto Rig Script. Part 1.**

he would cut and paste the corresponding script of the action. The end result is a compilation of every action preformed during a full rigging process.

The creation of the characters spine system is created, which was comprised of an inverse kinematics spline chain (IK) of joints. This created realistic and subtle joint movement in the torso of the body. An advanced leg system is constructed that relied on many IK chains to create the many movements a foot can perform. tools that make the foot tap, pivot, tip-toe, lift heel, are implemented as well as knee rotations. All of these tools exist on a single foot control handle that also translates the entire leg. The arm bones are then defined. Both IK and FK systems are implemented and controlled by a tool that switches between the two methods of control. FK systems are used to pose the arm more naturally to achieve free form movements. IK systems are used if the character has to plant his arm on a surface and still be able to pose the shoulder and elbow. Everyone one of these systems are translated into a script that describes the action of the production process.

In order for the automatic rig system to be universally used across many characters with widely different proportions, the rigging artist must be able to control where bones and joints are located for each figure. A template Maya file is written that contains joint locators and pre made control handles that the rigging artist can then use as a template starting point. The rigging artist must then import each unique character model into this premade maya scene. Each joint locator and control handle is named very specifically, for the auto rig script relies on these names

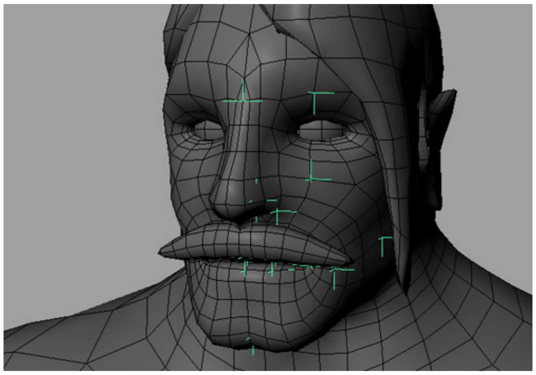
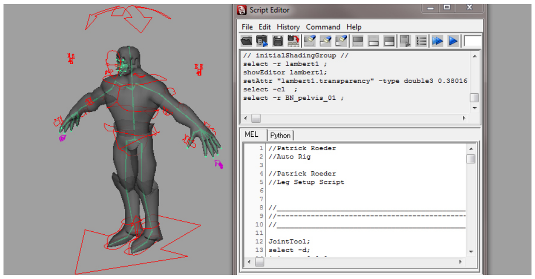


Figure 4.14: **1. Creation of an Auto Rig Script. Part 2.**



```
// initialShadingGroup //
select -r lambert1 ;
showEditor lambert1;
setAttr "lambert1.transparency" -type double3 0.38016
select -cl  ;
select -r BN_pelvis_01 ;
```

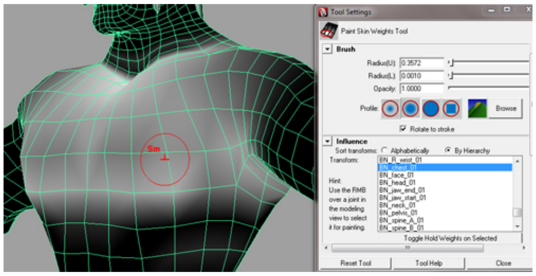
```
//Patrick Roeder
//Auto Rig

//Patrick Roeder
//Leg Setup Script


//_______________________________________________
//-----------------------------------------------
//_______________________________________________

JointTool;
select -d;
```

Figure 4.15: **1. Creation of an Auto Rig Script. Part 3.**

to implement its rigging actions. In figure 4.13, the figure Crimson Klausİ has been freshly imported into the template rigging scene. The character on the left shows what the rigging artist will see when initially loading the scene file. The figure on the right shows the completed work of positioning the pink joint locators and the blue control handles. Figure 4.14 shows an up close view of placing the face bone joint locators in their correction locations.

With all the initial setup work completed, all the rigging artist must do is load the script into the Maya's Script Editor and run the process. Figure 4.15 displays what the rigging artist will see after running the auto rig script; A complete body rig created for them in a few seconds.

Unfortunately, automating the weight painting process is not as easily done. After binding a skeleton to a character model, each vertex on the model is assigned


Tool Settings

Paint Skin Weights Tool

Brush

Radius(U): 0.3572

Radius(L): 0.0010

Opacity: 1.0000

☑ Rotate to stroke

Influence

Sort transforms: ○ Alphabetically ● By Hierarchy

Transform:

Hint: Use the RMB over a joint in the modeling view to select it for painting.

```
BN_R_wrist_01
BN_chest_01
BN_face_01
BN_head_01
BN_jaw_end_01
BN_jaw_start_01
BN_neck_01
BN_pelvis_01
BN_spine_A_01
BN_spine_B_01
```

Figure 4.16: **1. Weight Painting.** .

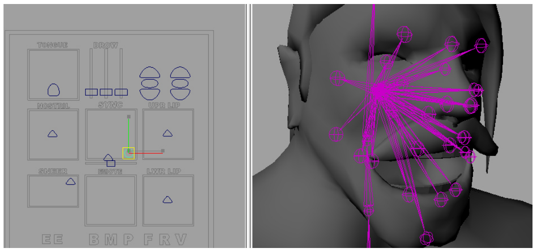


Figure 4.17: **1. Facial Rigging.** .

percentage values dictating how much influence the surrounding bones have. When a character is posed, bones will move, bringing along the vertices they influence by a percentage amount. The weight painting tools in Maya allow the artist to organically assign these values visually with a digital paintbrush workflow. The rigging artist must select a bone, then use grey scale feedback to paint the corresponding influence one vertex at a time. This process can be seen in figure 4.16.

A facial rig system is then constructed using only bones due to the limits of the game engine, Unity 3d. Each bone in the face is posed to create isolated facial movements. These movements include opening the mouth, smiling, closing the eyelids, and rotating the eyeballs. Set driven keys were then used to interpolate these movements into a visual interface used to control each of these movements separately. This system can be seen in figure 4.17.

4.5 Character Production: Animation

Animation in video games is the catalyst that brings their digital worlds alive. The process 3d animation consists of contorting 3d characters using a rigging system to create actions based on sequential key poses. An application, such as Maya, then interpolates the inbetween motion from key pose to key pose. With additional tweaking by the animator, this creates fluid movements for the game characters. Since these characters move in real time, a library of actions that a game engine uses in appropriate situations.

Since Beat Em Ups are combat heavy, a proper library of animations is necessary to convey the fast paced and brutal action of this genre. The same production cost issues occur in animation that occurs in the previous phases of character creation. With systems set up that allows for easy variation among characters that use the same rig, animation sets transfer over as well. These efficient shortcuts free up the animator to focus solely on creating animation libraries for the base character archetypes. Using proper animation methods and accurate video references the animation process can be as streamlined as the previous phases of production.

The first step in animation is to conceptualize the actions for each character. Considering the characters overall theme and attitude is very important in this step, because it is the animations that affect the player's perception of each character the most. If a character is fat and lazy, his/her movements must reflect this. The characters in a Beat Em Up, particularly the enemies, don't have many cinematic moments, so their story must be told in their actions. The conceptualization process starts with a list that describes various move sets and ends with ether filmed reference videos that act out each movement, or a collection of videos prerecorded for other projects that preform the same actions needed.

The most basic and fundamental animations needed for a character to exist comfortably in their world are now created. These animations include an idle,

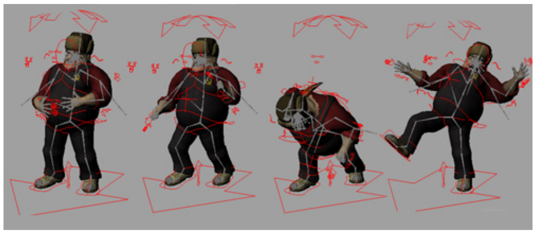
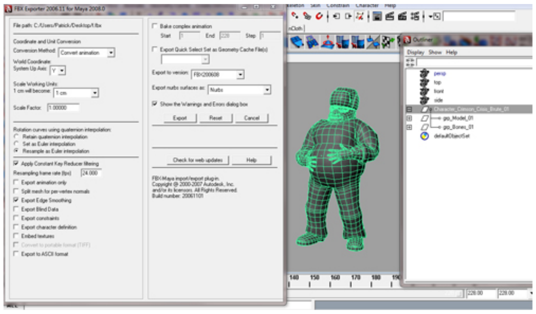


Figure 4.18: **1. Creating Basic Animations for each character Ex- Walk, Idle, Attack.** .



FBX Exporter 2006.11 for Maya 2008.0

File path: C:/Users/Patrick/Desktop/1.fbx

Coordinate and Unit Conversion
Conversion Method: Convert animation
World Coordinate System Up Axis: Y
Scale Working Units: 1 cm will become: 1 cm
Scale Factor: 1.00000

Rotation curves using quaternion interpolation:
○ Retain quaternion interpolation
○ Set as Euler interpolation
● Resample as Euler interpolation

☑ Apply Constant Key Reducer filtering
Resampling frame rate (fps) 24.000
□ Export animation only
□ Split mesh for per-vertex normals
☑ Export Edge Smoothing
□ Export Blind Data
□ Export constraints
□ Export character definition
□ Embed textures
□ Convert to portable format (TIFF)
□ Export to ASCII format

□ Bake complex animation
Start 1 End 228 Step 1
□ Export Quick Select Set as Geometry Cache File(s)
Export to version: FBX200608
Export nurbs surfaces as: Nurbs
☑ Show the Warnings and Errors dialog box

FBX-Maya import/export plug-in.
Copyright @ 2000-2007 Autodesk, Inc.
and/or its licensors. All Rights Reserved.
Build number: 20061101

Figure 4.19: **1. Exporting as an .fbx.**.

walk cycle, attack, animation loop of the character taking damage, and fall down animation. These five basic movements allow for base gameplay to exist. Key poses for these animations can be seen in figure 4.18.

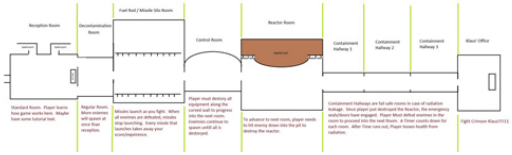
As gameplay features are created and added to the game, an animator can comfortably return to each character and add the necessary animation loops to them. After the animation process is complete, the model is optimized for exportation into Unity 3d, the game engine used for this game. This consists of making sure all face normal are pointing in the outward direction, deleting all non-deformation history on the mesh, and cleaning out other unnecessary items within the characters object hierarchy. The main bones that will be used in the game engine are selected and the character's animation is baked to every single keyframe on the time slider. This process ensures that every movement is captured on all the necessary bones.

After the animations are baked on the main bones used in the game engine, all extra items and objects can be deleted leaving only the game model and the game bones. The last step in character optimization for export is the actual act of exporting the model and bones out as a .fbx file type. The figure 4.19 shows the menu in Maya. This file type contains all pertinent information pertaining to texture UVs, polygons, and animation.

4.6 Environment Production

Environment production is the other half of video game art assets. While character production relies on a very particular series of steps and procedures, environment creation uses its very own unique production cycle. This phase of creation entails the building of buildings, outdoor settings, vegetation, household objects, vehicles, and any other models that exist in the game world. Modeling these assets can be vastly different from sculpting characters with the introduction of hard surface polygonal modeling. Rather than crafting organic meshes, hard surface modeling creates meshes that are very geometric and angular in shape.

The setting in a Beat Em Up is important to the player's enjoyment. Environments are one of the few elements that are constantly changing during the course of gameplay, and in a direct result, are one of the few elements that designers use to keep the player interested in their playtime. Environments also help tell the story of the Brawler. In the past, the Beat Em Up was very lite in the ways of cinematic production values, and the story usually relied on a short prologue. The various settings and progression from one locale to the next was usually all the background information a player got through the course of the game. To this degree, a logically progressing yet diverse set of environments is required for the success of a Beat Em Up game. This demand can put a strain on a small art team when creating these 3d art assets. Through a strong reliance on reference photographs, smartly designed level layouts, quick turnarounds on modeling and design iterations, and the use of



Reception Room
bathroom
bathroom
Decontamination Room
Fuel Rod / Missile Silo Room
Control Room
Reactor Room
reactor pit
Containment Hallway 1
Containment Hallway 2
Containment Hallway 3
Klaus' Office
Standard Room. Player learns how game works here. Maybe have some tutorial text.
Regular Room. More enemies will spawn at once than reception.
Missles launch as you fight. When all enemies are defeated, missles stop launching. Every missle that launches takes away your score/experience.
Player must destory all equipment along the curved wall to progress into the next room. Enemies continue to spawn untill all is destoryed.
To advance to next room, player needs to hit enemy down into the pit to destroy the reactor.
Containment Hallways are fail safe rooms in case of radiation leakage. Since player just destroyed the Reactor, the emergency seals/doors have engaged. Player Must defeat enemies in the room to proceed into the next Room. A Timer counts down for each room. After Time runs out, Player looses health from radiation.
Fight Crimson Klaus!!!!11

Figure 4.20: **1. Creating basic blueprint while designing gameplay elements related to the layout of the level.** .

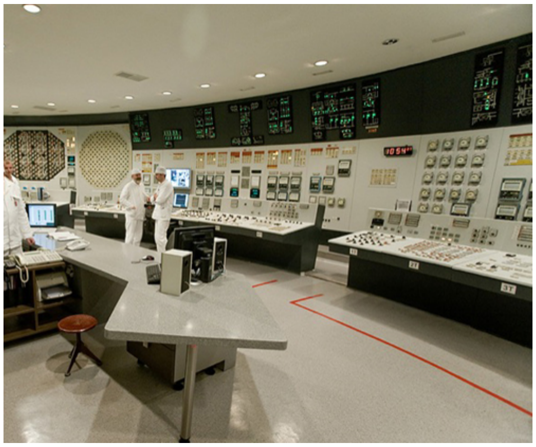


Figure 4.21: **1. Environment Reference Picture 1.** .

clever applications, A workflow is created allows for the creation of simplistic yet stylish environments that are resource friendly for a game engine.

The first step in developing the environment was to create a cohesive level theme. An abandoned nuclear power plant in soviet influenced Russia was choosen as the setting. A basic layout of the level that denotes room size, room shape, progression from one room to the next, individual room theme, and special environmental mechanics unique to each room is designed. The level blue print can be seen in Figure 4.20.

With an exact guide created, References were researchd. References to rooms such as a control room, a reactor room, various hallways, an office, and mechanical rooms were compiled. Figures 4.21 and 4.22 are a few examples of the photo references found.

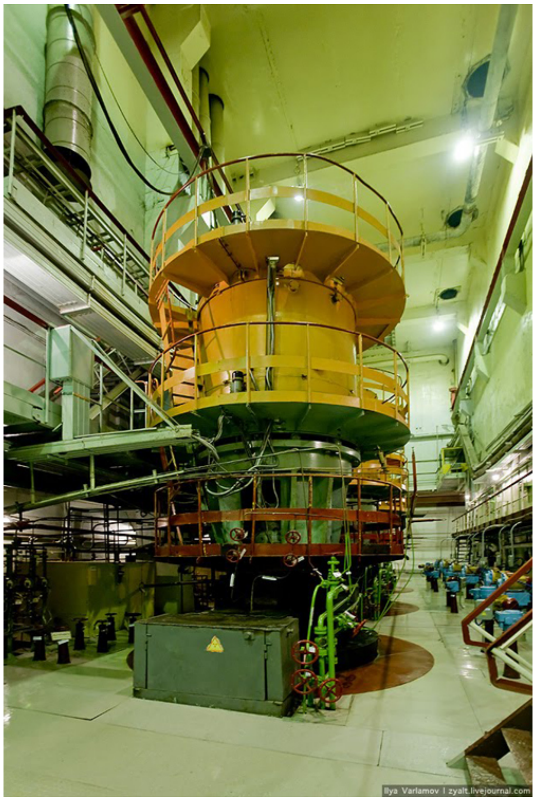


Figure 4.22: **1. Environment Reference Picture 2.** .

With a collection of reference photographs, a rough model of each room based off the 2d blueprint created earlier is constructed. Figure 4.23 displays a 3d version of the blueprint and foundation for the rooms to be created. With the first iteration of the basic low poly rooms built, a quick process of creating the UV's is completed. With clever overlapping of UV space, a repeating pattern can be used for surfaces such as walls and floors. This overlap technique as well as smart UV placement is used to maximize the UV space in an effort to get the most texture resolution for each room.

While the normal maps for each character were carefully sculpted in Zbrush for each character in the character production pipeline, this method would be too time consuming during environment creation. Instead of baking fine detail from high poly meshes onto the low poly game models, A program called Crazy Bump is used to

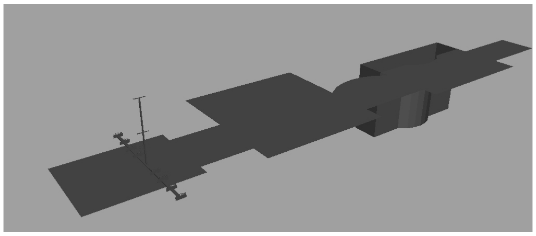


Figure 4.23: **1. Modeling basic meshes in accordance to the blueprints.** .

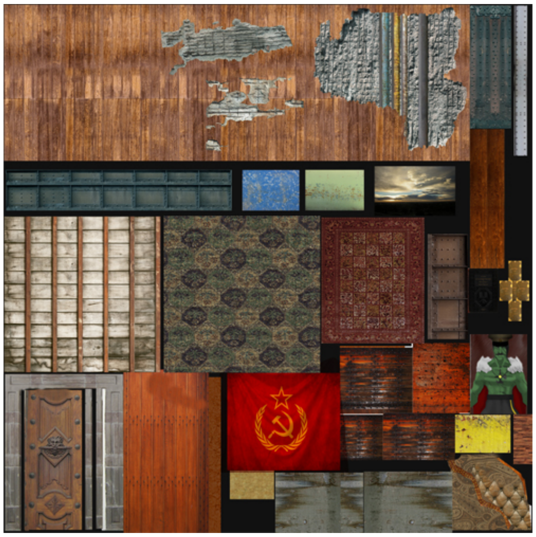


Figure 4.24: **1. A Complete Color Map Texture.** .

create normal maps from color photographs. Before this method can be used, each room must be textured completely. Using several photographs of various real world materials and objects, a collage of various images to paint a color map is created. A completed color map crafted in this way can be seen in figure 4.24.

Crazy Bump is a powerful texturing tool that creates high fidelity normal maps using a very fast workflow. It essentially analyzes a 2d photographic image and extracts a normal map from the color information. With simple tweaking and correct layering, Crazy Bump is extremely powerful. Various object textures are isolated in the color maps created for each room as seen in Figure 4.25. The layers within the objects texture is then further isolated in Figure 4.26, and exported into Crazy

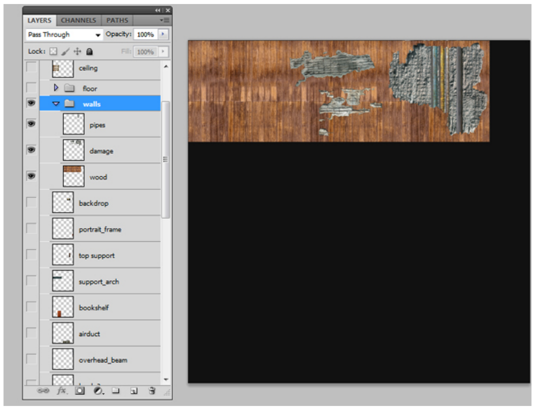


Figure 4.25: **1. Normal Map Creation Using Crazy Bump. Part 1** .

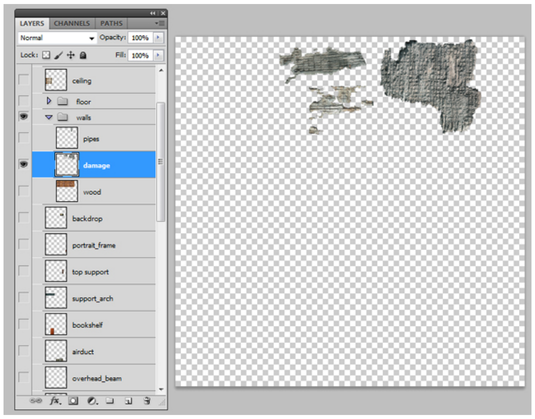
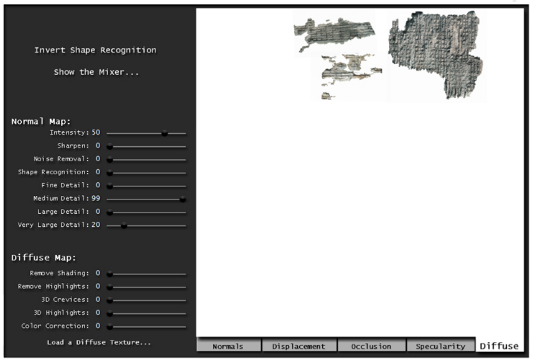


Figure 4.26: **1. Normal Map Creation Using Crazy Bump. Part 2** .

bump, as shown in Figure 4.27. Crazy Bump is then used to interpolate the color information into a RGB normal map. Various settings such as intensity of the normals, noise removal, and detail intensity can be adjusted to tailor the normal map to its best fidelity in Figure 4.28. After exporting the isolated normal map from Crazy Bump, the map is imported back into Photoshop as its own separate layer. This process for every single layer in the room's color map is then repeated. After all the normals have been calculated by Crazy Bump, Photoshop is used to stich each map together to create a fully realized normal map as seen in Figure 4.29.

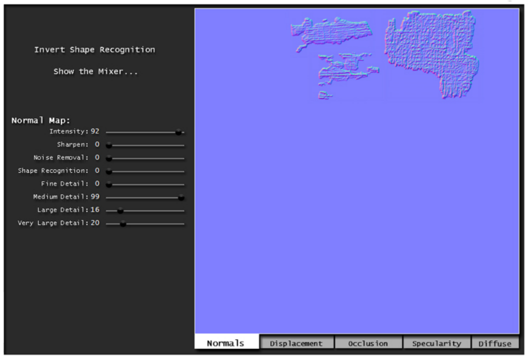
Throughout the modeling process, various series of iteration and design tweaks



Normal Map:
Intensity: 50
Sharpen: 0
Noise Removal: 0
Shape Recognition: 0
Fine Detail: 0
Medium Detail: 99
Large Detail: 0
Very Large Detail: 20

Diffuse Map:
Remove Shading: 0
Remove Highlights: 0
3D Crevices: 0
3D Highlights: 0
Color Correction: 0

Figure 4.27: **1. Normal Map Creation Using Crazy Bump. Part 3** .



Normal Map:
Intensity: 92
Sharpen: 0
Noise Removal: 0
Shape Recognition: 0
Fine Detail: 0
Medium Detail: 99
Large Detail: 16
Very Large Detail: 20

Figure 4.28: **1. Normal Map Creation Using Crazy Bump. Part 4** .

are made to organically improve the visuals of each room. These improvements include everything from additional prop creation, color texture improvements, UV space optimization, and subtle overall design cleanup. Figure 4.30 shows a first draft edition of the office room, while Figure 4.31 showcases a close to finish version of the same room. The use of photographic images and Crazy Bump normal creation allowed for swift turnarounds on these iterations, making this process quick and effective. Due to the lack of an environmental concept artist, this workflow became instrumental in the environment production process.

With the low poly game models, the color maps, and normal maps complete, creation of baked in light maps is next. The first step in the lighting process is the actual lighting of the space. This is done inside of Maya using various spot and area

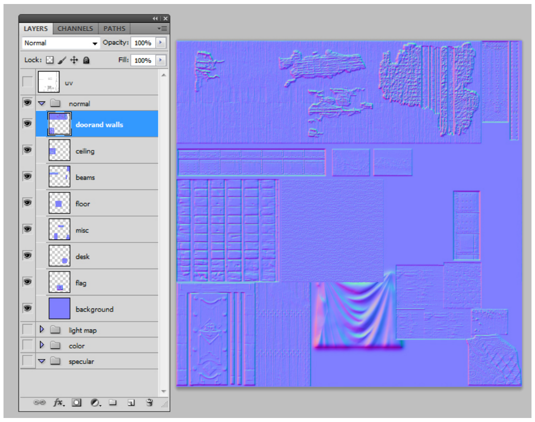


Figure 4.29: **1. Normal Map Creation Using Crazy Bump. Part 5** .

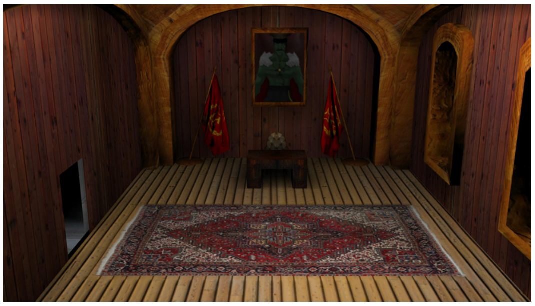


Figure 4.30: **1. Refining the design and textures of the Office Room. Old Design.** .

lights as displayed in Figure 4.32. Through a trial and error process to tweak the lights to the desired effect, a final lighting scheme is achieved as seen in Figure 4.33.

A problem arises when trying to transfer the lighting information from the pre-rendered visual effect to the low-poly real time game model. The process of baking lights consists of creating a texture map similar to a color map. Each surface must possess its own unique UV space since every surface receives light differently. A separate UV map must be created because of this since the original UV map shares UV space among several unique surfaces. The Figure 4.34 displays the UV map of the office room. This UV map is optimized to save room for larger resolution textures, and because of this, several polygonal faces share the same UV coordinates.

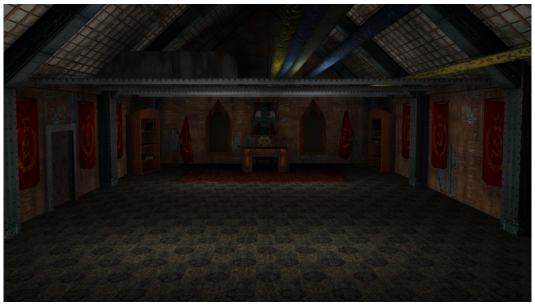
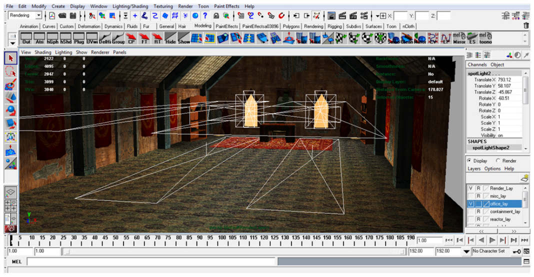


Figure 4.31: **1. Refining the design and textures of the Office Room. New Design.** .


File Edit Modify Create Display Window Lighting/Shading Texturing Render Toon Paint Effects Help
Rendering
Animation Curves Custom Deformation Dynamics Fluids Fur General Hair Modeling PaintEffects PaintEffectsa03896 Polygons Rendering Rigging Subdivs Surfaces Toon nCloth
Out Atri hGph hShd Plug UVwi DelHi Group CP FT RT Hide Show Mirror ES toono
View Shading Lighting Show Renderer Panels
Channels Object
Display Render
Layers Options Help
No Character Set

Verts: 2122 0 0
Edges: 4095 0 0
Faces: 2047 0 0
Tris: 3099 0 0
UVs: 3040 0 0

Backfaces: N/A
Smoothness: N/A
Instance: No
Display Layer: default
Distance From Camera: 178.027
Selected Objects: 15

spotLight2
Translate X 793.12
Translate Y 58.107
Translate Z -45.867
Rotate X -60.51
Rotate Y 0
Rotate Z 0
Scale X 1
Scale Y 1
Scale Z 1
Visibility on
SHAPES
spotLightShape2

V R Render_Lay
R misc_lay
V office_lay
R containment_lay
R reactor_lay

192.00

Figure 4.32: **1. Lighting the Office.** .

Figure 4.35 displays a second UV map that allows each polygonal face to exist in a unique UV space. Now that the game model contains two separate UV maps, the process of light baking can begin.

Maya is used to create an ambient occlusion map and a light source map. Figure 4.36 displays the menu used to create these baked light maps. The menu displays settings that allow for the bake of an ambient occlusion map can be seen in Figure 4.37. As noted previously, ambient occlusion is the shading process that describes the visuals of self-shadowing objects. After the ambient occlusion map is calculated and imported into Photoshop, the same is done for the light map itself in Figure 4.38. These maps are then combined in Photoshop to create a complete light map for use in the game engine in Figure 4.39. Light maps allow each room to be lit in a game engine for very low costs to the frame rate.

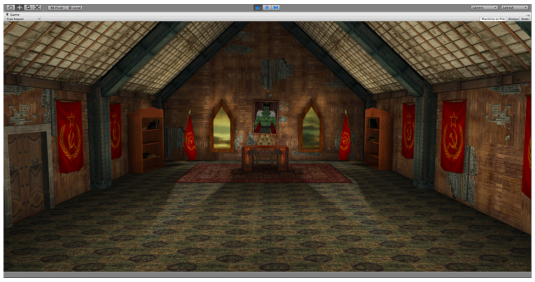


Figure 4.33: **1. Office Lit.** .

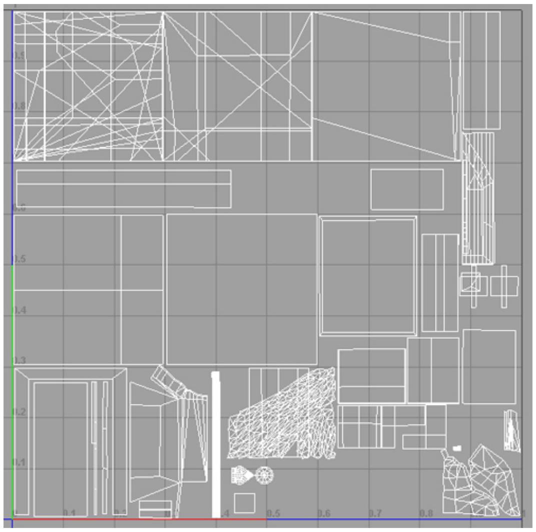


Figure 4.34: **1. Texture Map UV layout** .

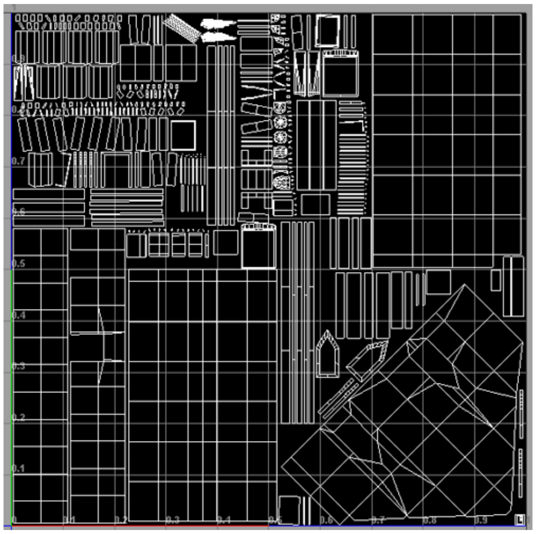
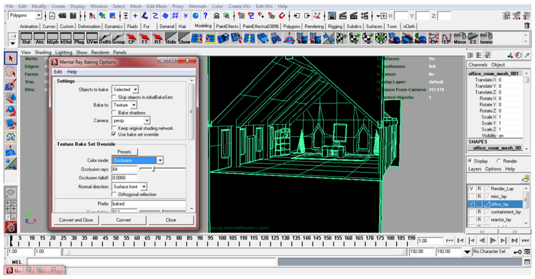


Figure 4.35: **1. Light Map UV layout**.



Mental Ray Baking Options

Settings

Objects to bake: Selected

□ Skip objects in initialBakeSets

Bake to: Texture

□ Bake shadows

Camera: persp

□ Keep original shading network

☑ Use bake set override

Texture Bake Set Override

Presets

Color mode: Occlusion

Occlusion rays: 64

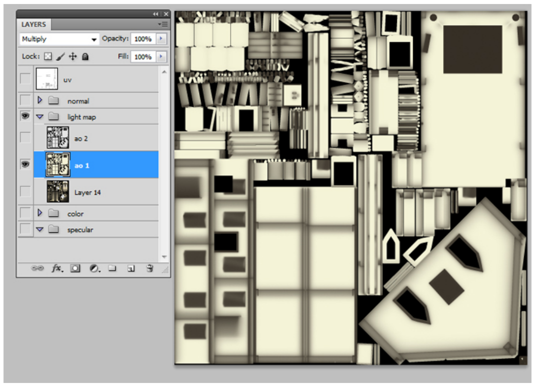
Occlusion falloff: 0.0000

Normal direction: Surface front

□ Orthogonal reflection

Prefix: baked

Figure 4.36: **1. Creating the Light Map. Step 1** .



LAYERS

Multiply Opacity: 100%

Lock: Fill: 100%

uv

normal

light map

ao 2

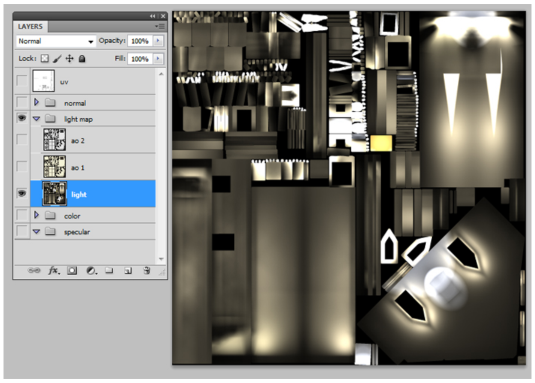
ao 1

Layer 14

color

specular

Figure 4.37: **1. Creating the Light Map. Step 2** .



Opacity: 100%

Fill: 100%

Figure 4.38: **1. Creating the Light Map. Step 3** .

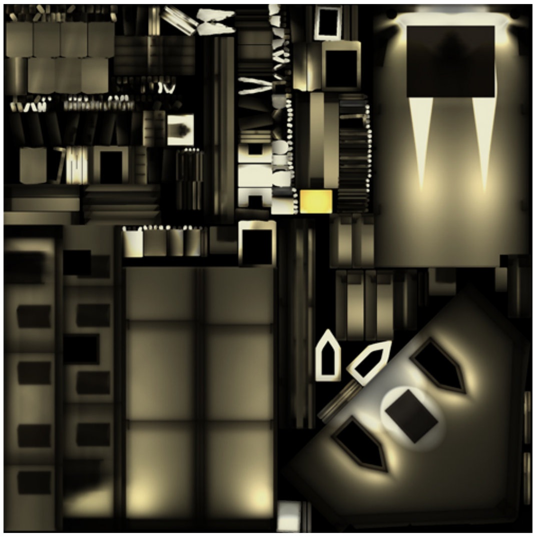


Figure 4.39: **1. Creating the Light Map. Step 4** .

Chapter 5

Results

In this chapter, results are provided that demonstrate the feasability of the approach to building a 3D video game in the Beat em up genre, with a small development team and a restrictive money and time budget. The game that is describe, titled "Terror Force 5", was developed by a team of three workers consisting of (1) a concept artist who created the character designs, (2) a programmer who worked with the game engine to create the gameplay, and (3) this author, an artist who created all 3d art assets. This effort resulted in the following contributions. The production issues and problems are examined that arise when translating a Beat Em Up video game from 2d to 3d. In order for a small team to be able to produce a finished product in a reasonable amount of time and with no monetary expenditure, a feasible production schedule must be developed and rigidly followed.

The schedule was made realizable by laying out a game concept and art direction that simplified the workload through stylization. All 3d art assets are created ranging from modeling characters and environments, to the rigging and animation of characters. And finally, a game level is created that applies the 3d art assets and the updated gameplay mechanics to a proof of concept game.

The production issues that occurred when Beat Em Ups were updated to 3d graphics provide the impetus and motivation for this work. The Beat em up genre was born from the 2d era, and when it was time for Beat Em ups to transfer over to the 3d movement, multiple issues arose. These issues and problems include various production pitfalls that many developers were confronted with during the late 90's. Art creation tools for 3d were very limiting, and 3d assets were very expensive in the game engines of the era. As a result, the art did not translate very well. Camera management was also another huge concern during this painful transition to 3d.

Not only was view obstruction a common issue, but the shift of perspective gave the genre an identity crisis due to the loss of its iconic “8 directions to move in 2d” mechanics. These production issues caused by the shift in the technology paradigm were also responsible for the lack of innovation and evolution of the genre’s gameplay. Beat Em Ups then grew stale, and lost the interest of their consumers.

After addressing the production issues described above, and taking into account the small size of the team, the game’s mechanics and art direction are created. In order for the Beat Em Up to be successful in the modern 3d era, I based the design around a few core tenets. The first being to shift the perspective back to the profile isometric view found in traditional Beat Em Ups. The second being to keep the combat as fluid and simplistic as possible. The third tenet is to create a production foundation that allowed for quick turnarounds, and rapid iteration and feature creation. And finally, a simplistic art style is designed that uses heavy stylization to keep the visual design interesting.

The last tenet is the most important in relation to keeping production feasible for a small team. In order to uphold the ideal of a simplistic and stylish art style, heavy inspiration is taken from the cartoons popular during the golden age of the Beat Em Up. These flat colored, pen and ink cartoons are a perfect example to from which to derive our style. They create the correct type of theme and mood for the game due to their nostalgic ties to the Era of the old Beat Em Up classics. They are also relatively easy to translate into 3d due to their simplistic and clean style. These cartoons were used to create the concept art for the game.

Experimentation sketches were presented to the prouction team to begin the visual design creation process. Figure 5.1 displays once such illustration. After a unified art direction for the characters was agreed upon, the concept artist created numerous sketches that were quick and loosely drawn. Figures 5.2 and 5.3 display two quickly produced concept iterations for 2 of the game’s characters. Sketching out these concepts allowed the 3d artist to give clearer direction to the concept artist

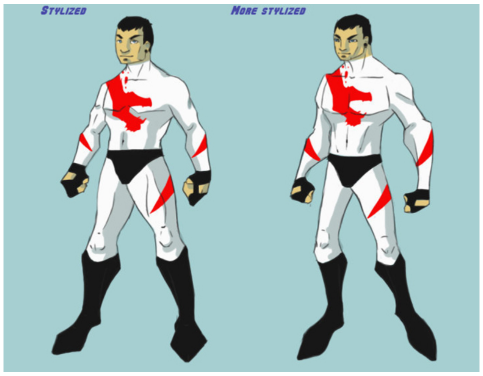


Figure 5.1: **Proportion Sketches Rising Sun**.

so he could create multiple designs in a short amount of time.

Final concept sketches were drawn, and a strict template is designed for the concept artist to follow. Figures 5.4 and 5.5 display finished character concept sheets that had their 3d models based on. These character concept sheets offer front and back views of the character with color. A side profile is included using only outlines to communicate the proportions of the character more easily. All three views of the character follow the same body size and proportions.

Using the blue print illustrations created by the concept artist, the character creation process can then begin. The first finished model is actually a template low poly model with correct polygon topology and average proportions. From this model, fully sculpted and textured models are produced. Figures 5.6 and 5.7 show finished character sculpts inside of Zbrush. These models are the high polygon density sculpts that all the texture maps are created from. The design blends highly detailed muscle systems and clothes details with the stylized proportions from the concept art.

Figures 5.8 and 5.9 display the finished models from the game. The figure illustrates the actual shaded character model, a wire frame detail of the model, and the various texture maps used in the final shaded characters.

After the modeling and texturing process is complete, each character is rigged

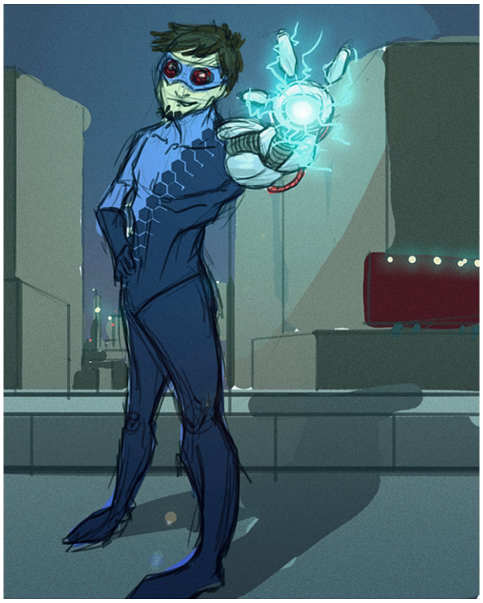


Figure 5.2: **Quick Sketch of Flux**.

and animated. Figure 5.10 showcases each character posed in their final states.

The next issue that is addressed is environment creation, which is produced for the proof of concept game. After setting the theme for the level, multiple photo references of Russian nuclear power plants, industrial machines, laboratories, and offices are gathered. These photo references serve as the blueprints for 3d asset creation. From these photographs, the low polygon models that would be used in game are made and later textureed. After modeling and texturing was completed, the rooms are lit with multiple light sources. Figures 5.11, 5.12, and 5.13 showcase pre-rendered screenshots of a few of the rooms from the level.

These pre-rendered screens are used to gauge the overall aesthetic of each room before beginning the optimization processes to transport them into the game engine. The results of these preparations are the actual in game environments illustrarted in Figures 5.14, 5.15, and 5.16 These screenshots are taken from the game as seen in real time rendering and showcase the perspective from which the player views the action. The end result of all these contributions is a short 9 room level featuring 2

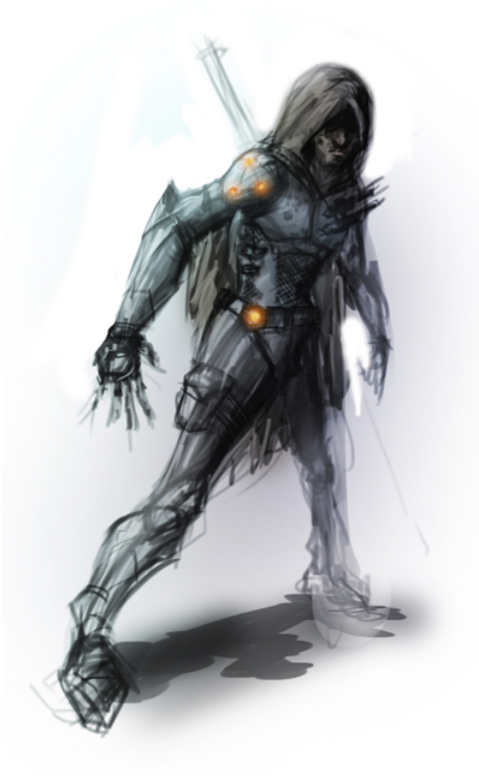


Figure 5.3: **Quick Sketch of Shroud**.

playable characters, and 3 non playable enemies. The game is “Terror Force 5” and revolves around a band of heroes styled after many of the fictional and iconic heroes groups of the 80’s. Through the implementation of 3d techniques that recapture the essence of the 2d gameplay of traditional Beat Em Ups, a low budget proof of concept game is crafted that proves that Beat Em Ups are not only feasible in the modern era, but relevant to the consumer as well.

This production was presented in Atlanta at the Southern Interactive Entertainment and Game Expo on October 5th of 2012 (www.siegecon.net). The game engine Terror Force 5 is built on, Unity 3d, was the one of the subjects discussed on a panel of other 3d artists and programmers. Topics such as Terror Force 5 character optimization and exportation were discussed in detail, as well as other game engine related matters. The entire production pipeline of Terror Force 5 will also be

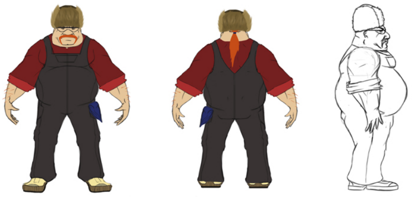


Figure 5.4: **Brute Character Concept Sheet**.

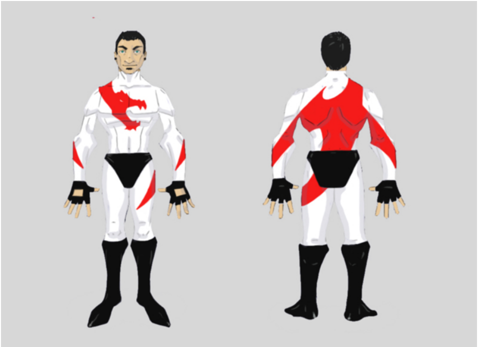


Figure 5.5: **Rising Sun Character Concept Sheet**.

the topic of a lecture at the East Coast Game Conference in Raleigh NC on April 26th of 2012 (www.ecgconf.com). The art development session will outline the entire production cycle of the game while touching on tips, tricks, and warnings of pitfalls that can occur during game creation. An emphasis will be placed on creating the right kind of game for a small team.

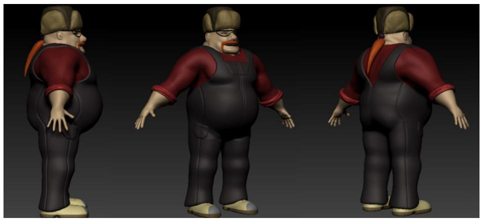


Figure 5.6: **Brute ZBrush Sculpt**

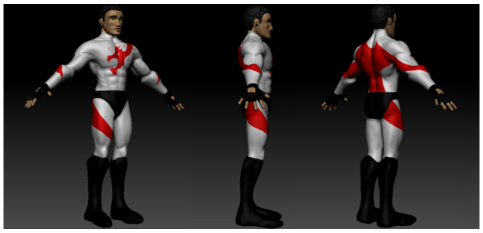


Figure 5.7: **Rising Sun Character Concept Sheet**.

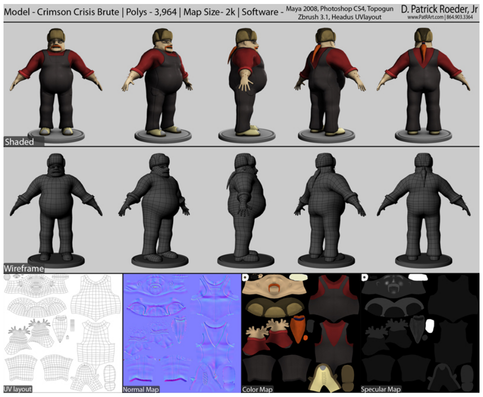

Model - Crimson Crisis Brute | Polys - 3,964 | Map Size- 2k | Software - Maya 2008, Photoshop CS4, Topogun Zbrush 3.1, Headus UVlayout
D. Patrick Roeder, Jr
www.PatRArt.com | 864.903.3364
Shaded
Wireframe
UV layout
Normal Map
Color Map
Specular Map

Figure 5.8: **Brute Game Model**.

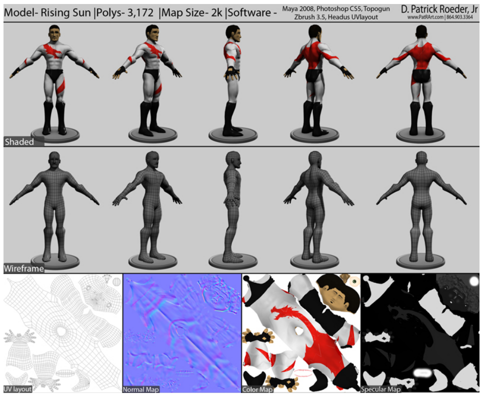

Model- Rising Sun |Polys- 3,172 |Map Size- 2k |Software -
Maya 2008, Photoshop CS5, Topogun
Zbrush 3.5, Headus UVlayout
D. Patrick Roeder, Jr
www.PatRArt.com | 864.903.3364
Shaded
Wireframe
UV layout
Normal Map
Color Map
Specular Map

Figure 5.9: **Rising Game Model**.

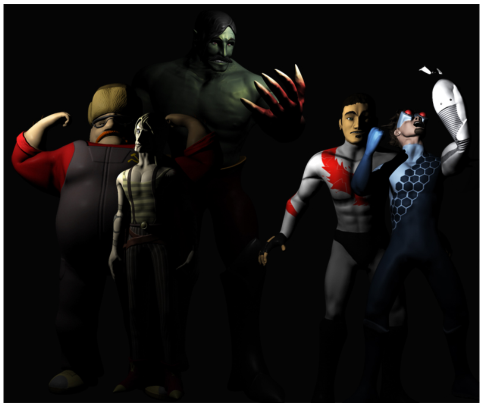


Figure 5.10: **The Characters of Terror Force 5 Posed.**.

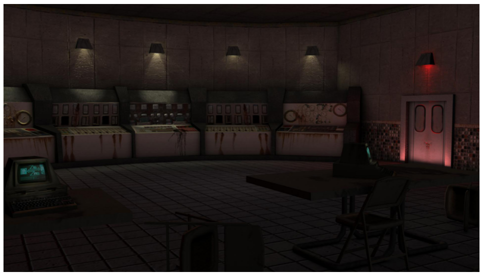


Figure 5.11: **Control Room Pre-Rendered**.

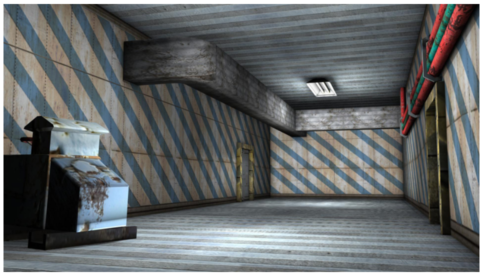


Figure 5.12: **Containment Room Pre-Rendered**.

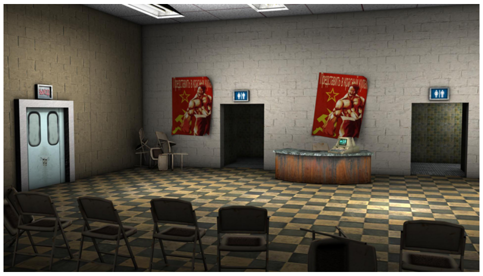


Figure 5.13: **Reception Room Pre-Rendered**.

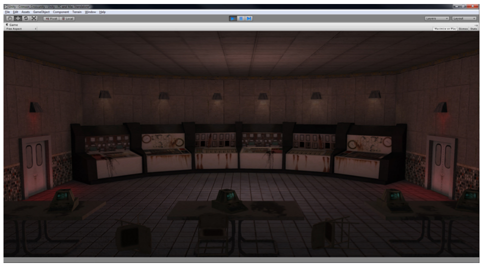


Figure 5.14: **Control Room In Game Screenshot**.

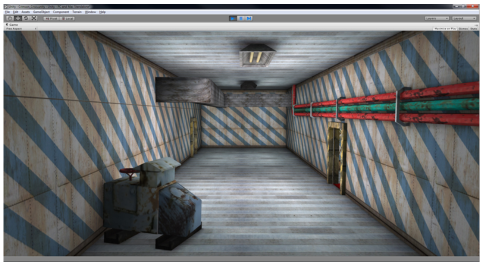


Figure 5.15: **Containment Room In Game Screenshot**.

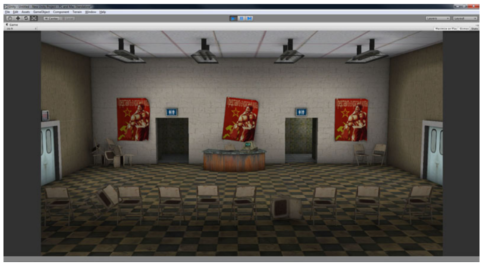


Figure 5.16: **Reception Room In Game Screenshot**.

Chapter 6

Conclusions

In this thesis, we have described our design, production and implementation of a 3D video game in the Beat em up genre. We have designed and implemented this game with a small team of three developers. We have demonstrated that the production of a Beat Em Up game is feasible with a small team using 3D technology.

Bibliography

[1] Early Beat Em Up. http://www.racketboy.com/retro/beatemups/2010/08/beatem-ups-101-all-you-need-to-know-about-brawlers.html.

[2] Gamespy. http://wii.gamespy.com/articles/103/1039605p1.html.

[3] Good Games. http://www.abc.net.au/tv/goodgame/stories/s2067970.htm.

[4] The co-op beat-em-up. http://www.co-optimus.com/editorial/131/page/4/the-co-op-beat-em-up-retrospective.html.

[5] Tomb Raider. http://en.wikipedia.org/wiki/Core_Design.